NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A CONCEPT FOR CONTINUOUS MONITORING THAT REDUCES REDUNDANCY IN INFORMATION ASSURANCE PROCESSES

by

Sophia Kostopoulos

September 2011

Thesis Advisor:	Karen Burke
Second Reader:	George Dinolt

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704–0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704–0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2011	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE A Concept for Continuous Monitoring that Reduces Redundancy in Information Assurance Processes		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Kostopoulos, Sophia			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943–5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol No. N/A.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

This thesis analyzes the structure of a few of the Information Assurance (IA) processes currently being used in the United States Government. The general structure of these processes is uncovered and used to create a Continuous Monitoring Process that can be used to create a tool to incorporate any process of similar structure. A proof-of-concept application is drafted to demonstrate the main aspects of the proposed tool. The possibilities and implications of the proof-of-concept application are explored, including the future work required to develop a fully functional and automated version of the proposed Continuous Monitoring tool.

14. SUBJECT TERMS Information Assurance, Certification and Accreditation (C&A), Continuous Monitoring, DIACAP.			**15. NUMBER OF PAGES** 119
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01–280–5500 Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A CONCEPT FOR CONTINUOUS MONITORING THAT REDUCES REDUNDANCY IN INFORMATION ASSURANCE PROCESSES

Sophia Kostopoulos
Civilian, Federal Cyber Corps
B.Eng., University of Kent, England, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2011

Author: Sophia Kostopoulos

Approved by: Karen L. Burke
Thesis Advisor

George W. Dinolt
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis analyzes the structure of a few of the Information Assurance (IA) processes currently being used in the United States government. The general structure of these processes is uncovered and used to create a Continuous Monitoring Process that can be used to create a tool to incorporate any process of similar structure. A proof-of-concept application is drafted to demonstrate the main aspects of the proposed tool. The possibilities and implications of the proof-of-concept application are explored, including the future work required to develop a fully functional and automated version of the proposed Continuous Monitoring tool.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ATC	Authorization to Connect
ATO	Authorization to Operate
C&A	Certification and Accreditation
CA	Certification Authority
CAP	Connection Approval Process
CCRI	Command Cyber Readiness Inspection
CJCSI	Chairman of the Joint Chiefs of Staff Instruction
CL	Confidentiality Level
CND	Computer Network Defense
DAA	Designated Accrediting Authority
DATO	Denial of Authorization to Operate
DIACAP	DoD Information Assurance Certification and Accreditation Process
DIP	DIACAP Implementation Plan
DISA	Defense Information Systems Agency
DISN	Defense Information System Network
DoD	Department of Defense
DoDD	DoD Directive
DoDI	DoD Instruction
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
FSO	Field Security Operations
GIG	Global Information Grid

HTML	Hypertext Markup Language
IA	Information Assurance
IATO	Interim Authorization to Operate
IATT	Interim Authorization to Test
IS	Information System
JSP	JavaServer Pages
KS	Knowledge Service
MAC	Mission Assurance Category
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
POA&M	Plan of Action and Milestones
RMF	Risk Management Framework
SCAP	Security Content Automation Protocol
SIP	System Identification Profile
SOP	Standard Operating Procedure
SP	Special Publication
SQL	Structured Query Language
SRR	Security Readiness Review
STIG	Security Technical Implementation Guide

ACKNOWLEDGMENTS

I am grateful to the National Science Foundation for the opportunities presented to me through the Scholarship for Service (SFS) Program. This material is based on work supported by the National Science Foundation under Grant DUE-0414102. To Professor Cynthia Irvine, the Naval Postgraduate School SFS Principal Investigator, it has been an honor to learn from you. A special thanks to Valerie Linhoff for all the behind the scenes SFS work that allowed me to concentrate on my classes and this thesis. Professor Arijit Das, thank you for taking time to help me with some of the technical difficulties of the proof-of-concept. To my advisors, Professor Karen Burke and Professor George Dinolt, thank you for your support throughout the thesis and job hunting. You are all an inspiration.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In international relations, offensive advantage “means that it is easier to destroy the other’s army and take its territory than it is to defend one’s own” [1]. This can be translated in terms of cyber security to mean that it is easier to destroy the availability of the other’s information infrastructure and take its confidential information than it is to defend one’s own information infrastructure. Due to the fact that there is a clear offensive advantage in cyber warfare, it is important to ensure the security of information systems by having information assurance security controls in place and up-to-date. Information Assurance (IA) consists of the “measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation” [2]. Security controls are “the management, operational and technical controls (e.g., safeguards or countermeasures) prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information” [2].

Title III of the E-Government Act, referred to as the Federal Information Security Management Act (FISMA), requires federal agencies to provide security for the information and information systems that support the organization, and to conduct annual agency program reviews. The requirements of FISMA include developing, documenting, and implementing an information security program and developing and maintaining an inventory of information

systems under the control of the organization. Personnel must be trained to assist in complying with the required policies and senior leaders must provide information security for assets under their control. The key requirements are to provide information security protections commensurate with the assessed risk and to compose annual reports on the effectiveness of the organization's information security program [3].

Agencies are required, by the OMB Circular A-130 Appendix III, to review the security controls of their information systems to ensure that changes do not have a significant impact on security, IA controls continue to perform as intended, and security plans remain effective. The OMB Circular A-130 also requires Federal information systems to include a minimum set of controls and be certified and accredited [4].

The DoD Instruction 8510.01 *DoD Information Assurance Certification and Accreditation Process (DIACAP)* is how the OMB and FISMA requirements are met. The DIACAP ensures the risks associated with the information system (IS) are acceptable. The DIACAP checks for compliance against the IA controls in the DoD Instruction 8500.2 *Information Assurance (IA) Implementation*. Other IA processes are conducted throughout the system life cycle and vary depending on the department, organization, or service. Some ISs fall under more than one category, or process, and are required to be checked for each process. This requires more time and effort while causing unnecessary redundancy. This redundancy can be reduced through continuous monitoring and reuse of automated scans and manual checks of the IA

controls. Vulnerabilities to the IS can occur if IA controls are not performing as intended or new weaknesses to the system are not addressed. Without continuous monitoring, these vulnerabilities may go unnoticed until DIACAP re-certification which may be years away [5, 6].

A. RESEARCH DISCUSSION

There are several IA Processes currently being used throughout the United States Government. Each department, such as the Department of Defense (DoD) and Department of State, has its own processes and internal standard operating procedures (SOPs). Even within the DoD, each service, agency, and organization implements the processes differently or has created their own version of the processes. As a result, the same IA controls are checked in several processes, creating redundant work and wasting critical time.

This thesis defines a concept of continuous monitoring that attempts to create a process from the similar structure of several existing IA processes. The specific documents and procedures that differ among the processes can be incorporated to reuse scan results and manual checks that have already been conducted on an IS. This concept is demonstrated by means of a proof-of-concept application that demonstrates the common structure of the IA processes and conveys the potential for a fully functional automated Continuous Monitoring tool that can implement any IA process with the mentioned structure. The continuous assessment of the security controls will ensure the IA posture is maintained and offers timely mitigation of vulnerabilities so that ISs are better defended.

B. SCOPE

The scope of the thesis is the concept of a continuous monitoring process that encompasses existing IA processes with similar structures. This research demonstrates the concept with the creation of a proof-of-concept application and is not meant to create a fully functional tool. If this concept is to be adopted, additions to the tool would be necessary as discussed in Chapter V.

C. ORGANIZATION OF THESIS

Background information on a few of the IA processes is presented in Chapter II. Chapter III reveals the redundancy between the discussed processes and uses the similar underlying structure to design a continuous monitoring process. The proof-of-concept application is described in Chapter IV and mapped back to the continuous monitoring process of Chapter III. Chapter V discusses the implications of the proof-of-concept tool and the future research that is required to make a fully functional and automated version of the tool.

II. BACKGROUND

A. DIACAP

The DoDI 8510.01 [5] establishes a process for DoD IA Certification and Accreditation that will authorize the operation of DoD information systems in accordance with FISMA [3], DoDD 8500.01 *Information Assurance* [7], DoDI 8500.2 *Information Assurance Implementation* [6], and DoDD 8100.1 *Global Information Grid (GIG) Overarching Policy* [8]. The process, shown in Figure 1 [9], consists of five activities that manage the implementation of IA controls and provide visibility of accreditation decisions regarding the operation of DoD information systems.

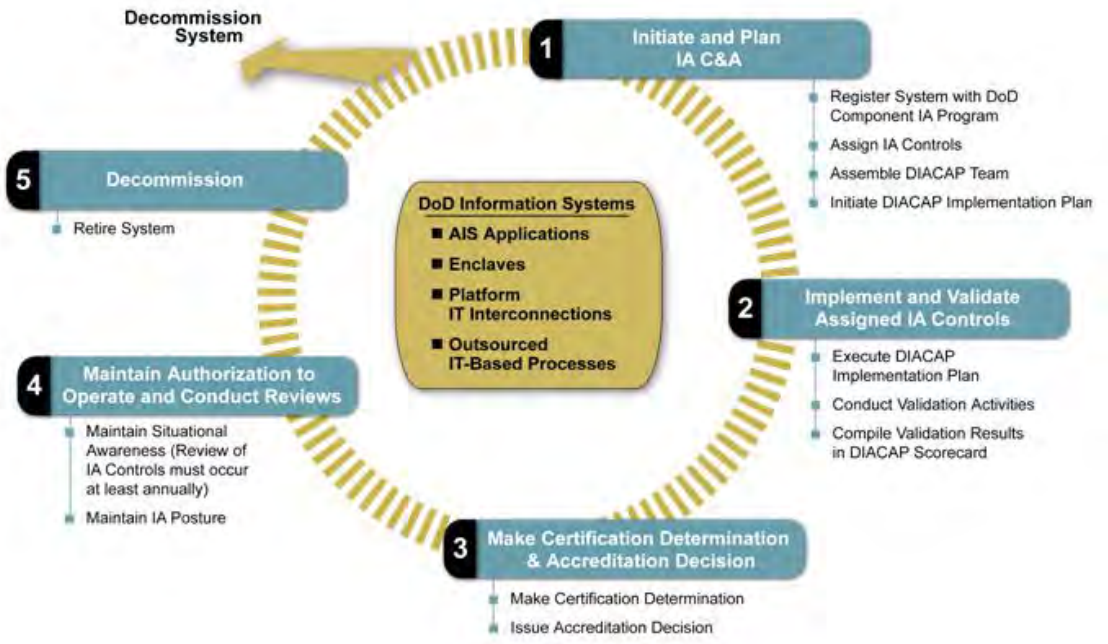


Figure 1. DIACAP Activities

1. Initiate and Plan IA C&A

The first activity consists of preparatory actions for IA Certification and Accreditation. The Information System Type is determined and the system is categorized with a Mission Assurance Category (MAC) and Confidentiality Level (CL) as defined in the DoDD 8500.01. The System Identification Profile (SIP) is developed and the system is registered with the DoD Component IA Program and other organization-specific registration tasks are performed. The baseline IA controls are generated from the DoDI 8500.2 based on the type and category of the IS. These baseline controls are adjusted to account for inherited, not applicable, and system-specific controls, and then compiled in the IA Control Implementation Plan.

The Certification and Accreditation (C&A) Plan is formed from the IA Control Implementation Plan, and Validation Plan and Procedures. The DIACAP Implementation Plan (DIP) contains the assigned IA controls, their estimated completion date, implementation status, responsible entities, resources, architecture, and technical details. The DIP is reviewed and approved once the DIACAP team is in agreement on the security requirements and schedule. The DIACAP team is assembled to initiate the C&A Plan and the DIP.

2. Implement and Validate Assigned IA Controls

In the second activity, the DIP is executed and the assigned IA controls are implemented. Other systems are also checked in order to verify inherited controls. The implementation is documented and the DIP is updated. Validation activities are conducted to assess the

effectiveness of the IA controls. Implementation and validation guidelines are available at the DIACAP Knowledge Service [10]. The compliance status from the Validation Report is recorded in the DIACAP Scorecard, and, if corrective actions are necessary, the Plan of Actions and Milestones (POA&M) is prepared and/or updated. The POA&M identifies controls that are non-compliant, the tasks that need to be accomplished, and the scheduled completion date for each task. The non-compliant controls are categorized using the DIACAP Severity Codes. The code indicates the risk level of the control, the likelihood of consequences due to non-compliance, and the urgency required for corrective actions. Non-compliant controls should be prioritized for remediation based on the impact codes within each severity category.

3. Make Certification Determination and Accreditation Decision

The certification of an information system is the

> … comprehensive evaluation of the technical and non-technical security safeguards of an information system to support the accreditation process that establishes the extent to which a particular design and implementation meets a set of specified security requirements. [2]

The certification determination and accreditation decision takes place in activity three.

The Certification Authority (CA) makes the certification determination based on the actual validation results, the impact codes and severity categories of non-compliant controls, expected exposure time, and costs of mitigation. The CA forwards either the Executive or

Comprehensive Package to the Designated Accrediting Authority (DAA) to issue an accreditation decision. The Executive Package consists of the System Identification Profile (SIP), DIACAP Scorecard, and POA&M if required. The Comprehensive Package includes the documents of the Executive Package, as well as the DIACAP Implementation Plan (DIP) and Certification documentation. Accreditation is a declaration

> … that an information system is approved to operate at an acceptable level of risk, based on the implementation of an approved set of technical, managerial, and procedural safeguards. [2]

The DAA reviews the package and assesses the residual risk. If it is acceptable, the DAA issues the accreditation decision (i.e. Authorization to Operate (ATO), Interim Authorization to Operate (IATO), or Interim Authorization to Test (IATT)) and assigns an Authorization Termination Date on the DIACAP Scorecard. If the risk is unacceptable, a Denial of Authorization to Operate (DATO) will be issued.

4. Maintain Authorization to Operate and Conduct Reviews

In this activity, the DIACAP team works to maintain the Authorization to Operate (ATO) through the sustainment of an acceptable IA posture. This activity initiates and updates a Life cycle Implementation Plan for the IA controls that continuously monitors the system and assesses the quality of the IA controls. A System Monitoring Program is developed to maintain situational awareness. Action is taken on any Information Assurance Vulnerability Alerts. Performance reviews are conducted annually, as required by

FISMA, and re-accreditation is initiated every three years, as required by OMB Circular A-130. Testing is conducted on a select number of IA controls and the results are given to the DAA and CA with the annual performance review. “The results of an annual review or a major change in the IA posture at any time may also indicate the need for recertification of the IS” [5]. This will generate changes to the DIACAP Package, a written Statement of IA Controls Review, and an updated Accreditation Decision.

5. Decommission

“Prior to decommissioning, any inheritance relationships should be reviewed and assessed for impact” [5]. This activity reviews inheritance relationships to ensure the system’s removal from operation does not negatively affect the operation of associated systems. The DIACAP registration information and system-related data are disposed of or updated to reflect the retiring of the system. The IS is then uninstalled or disconnected. A Denial of Authorization to operate is issued by the DAA and the system may no longer operate.

B. RISK MANAGEMENT FRAMEWORK

The NIST SP 800–37, *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach* [11], defines the Risk Management Framework (RMF). The RMF

> … incorporates the FISMA- and OMB-related security standards and guidance to provide a holistic solution for managing risk to an organization’s information and information systems. [12]

The RMF allows for situational awareness through the constant analysis of the IA posture of the information systems. This knowledge helps in the swift detection and mitigation of problems and vulnerabilities.

The objective of continuous monitoring is to determine if security controls in an information system (IS) continue to be effective over time despite inevitable changes that occur in the system and environment in which the system operates. The six steps of the RMF are illustrated in Figure 2 [11]. The RMF helps reflect the current status of security programs and controls to protect the confidentiality, integrity, and availability of systems and make informed judgments that appropriately mitigate risk to an acceptable level.

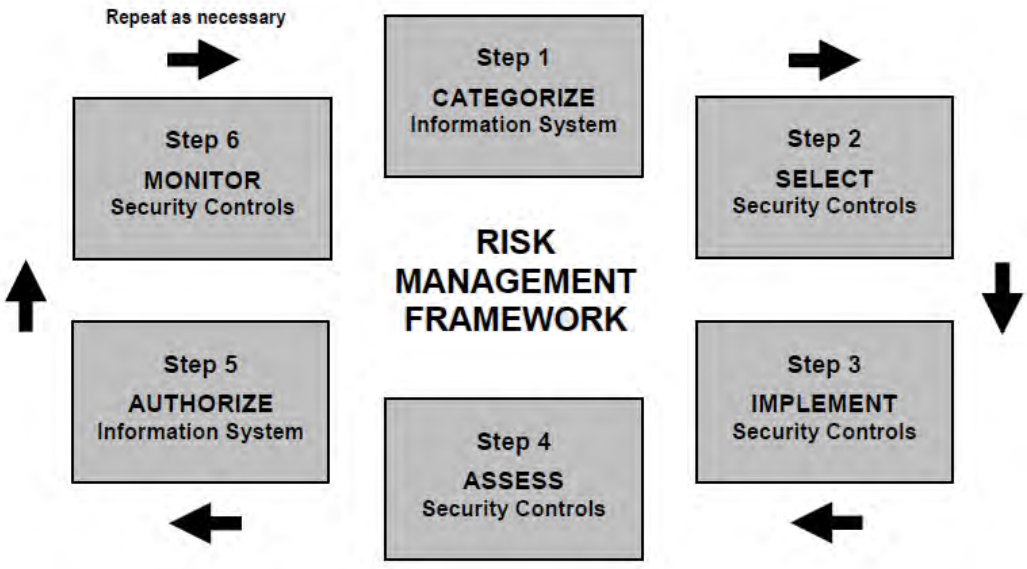


Figure 2. NIST Risk Management Framework

1. Categorize Information System

The first step in the Risk Management Framework is to categorize the information system.

> Security categorization determinations consider potential adverse impacts to organizational operations, organizational assets, individuals, other organizations, and the Nation. [11]

The first task uses FIPS 199, *Standards for Security Categorization of Federal Information and Information Systems* [13], and NIST SP 800–60, *Guide for Mapping Types of Information and Information Systems to Security Categories* [14] to categorize the IS. The second task in this step is to document the system description in the Security Plan along with the results of the security categorization. On page 21 and 22 of the NIST SP 800–37 [11], there are a few examples of what information the system description would include. The level of detail provided in the description should be commensurate with the security categorization of the IS. In the final task of this step, the IS is registered with the appropriate organizational offices in order to inform the IS owner/manager of the system's existence, its key characteristics, and the security implications for the organization. This "provides an effective management/tracking tool that is necessary for security status reporting" [11].

2. Select Security Controls

The second step involves selecting an initial set of security controls for the IS to reduce threats and manage risks. The first task is to identify the common, or

inherited, security controls that are provided by the organization. The IS owner should ensure the inherited controls deliver sufficient protection and that they are made aware of any changes to the status of the inherited controls that may adversely affect the IS. These common controls are documented in the Security Plan.

The FIPS 200, *Minimum Security Requirements for Federal Information and Information Systems* [15], requires organizations to meet the minimum security requirements by selecting the appropriate security controls and assurance requirements based on the security categorization of the IS and the NIST SP 800–53, *Recommended Security Controls for Federal Information Systems and Organizations* [16]. The baseline security controls are selected in task two, and then tailored and supplemented as needed based on the organizational assessment of risk. These controls and their intended application are documented in the Security Plan.

The Continuous Monitoring Strategy is developed in the third task to monitor the effectiveness of the security controls and to identify any changes to the IS and its environment of operation. The strategy identifies the security controls to be monitored, and the method and frequency of the analyses. Controls that are volatile, critical to protection, or identified in the POA&M are selected for monitoring. They are “assessed as frequently as necessary consistent with the criticality of the function and capability of the monitoring tools” [11]. The strategy also defines the recipients of status reports and how to monitor changes to the system.

The final task in this step is for the authorizing officials to review and approve the Continuous Monitoring Strategy and Security Plan. If the Security Plan is deemed unacceptable, the plan is sent back for appropriate action. If the plan is deemed acceptable, the authorizing official is agreeing to the set of security controls proposed to meet the security requirements for the IS and the residual risk associated with implementing these controls as intended. The approval of the Security Plan allows the process to proceed to the next step.

3. Implement Security Controls

Implementing the security controls specified in the Security Plan is the first task in step three. The second task is to update the Security Plan with the security control documentation which allows for traceability of decisions. The functional description of the implementation of the security controls should state how the controls are employed within the IS and its environment of operation, including planned inputs, expected behavior, and expected outputs.

4. Assess Security Controls

The effectiveness of the security controls is assessed in step four. The first task is to develop a Security Assessment Plan that will place a bound on the level of effort put into the assessment. NIST SP 800–53A, *Guide for Assessing the Security Controls in Federal Information Systems and Organizations* [17], provides guidance for building effective Security Assessment Plans. The plan should identify the type of assessment (i.e. audit,

continuous monitoring, certification, etc.), the objectives for the assessment, and the detailed procedures for conducting the assessment. The plan is then reviewed and approved by the appropriate officials. The second task is to execute the plan and assess the security controls to

> … determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the information system. [11]

The assessment is conducted in accordance with the procedures in the Security Assessment Plan.

The results of the assessment include recommendations on how to correct non-compliant controls and reduce or eliminate identified vulnerabilities. They are compiled in the Security Assessment Report during the third task. The report documents the issues, findings, and recommendations of the assessment. The fourth task uses the results and the report to conduct initial remediation actions based on the findings and recommendations. Security controls are reassessed after remediation, as appropriate. The Security Plan is updated to include the findings and actions resulting in this step.

5. Authorize Information System

The authorization decision for the IS is made in step five. The first task uses the information in the Security Assessment Report to prepare the POA&M. In task two, the Security Authorization Package is compiled and submitted to the authorizing official for adjudication. The package consists of the Security Plan, Security Assessment Report,

and POA&M, which collectively depict the current security state of the information system and current risk posture.

The third task is to determine the risk to organizational operations, organizational assets, individuals, other organizations, or the Nation based on risk assessments and the Security Authorization Package. The authorizing official utilizes the information provided and balances security considerations with mission and operational needs to make an authorization decision in task four of this step. The Authorization Decision document consists of the authorization decision, the terms and conditions for authorization, and the authorization termination date. If the authorizing official concludes that the risk is acceptable, the system will receive authorization to operate, otherwise, the system will not be authorized to operate [11].

6. Monitor Security Controls

The final step of the Risk Management Framework is to monitor and assess the security controls in the information system on a continuous basis to ensure and demonstrate security due diligence. This step includes the typical activities of continuous monitoring: updating documents, conducting security impact analyses, reporting security status of the system, and conducting ongoing security control assessments and risk determinations. These activities allow the authorizing officials to manage risk and maintain the security authorization over time.

The first task is to determine the security impact of information system and environment changes. The second task is to conduct ongoing security control assessment by

selecting a subset of the security controls in accordance with the previously defined monitoring strategy. The third task is to conduct ongoing monitoring activities, assessment of risk, and remediation of outstanding items in the POA&M. Task four is to update the components of the Security Authorization Package based on the results of the previous tasks. This will facilitate near real-time situational awareness and management.

In line with the monitoring strategy, the fifth task is to report the security status of the information system to the appropriate organizational officials on an ongoing basis. The sixth task is to conduct ongoing risk determination and acceptance by reviewing the reported security status of the IS on an ongoing basis to determine whether the risk remains acceptable. The final task is performed when an information system is removed from service. In this task, the decommissioning strategy is implemented and required actions are taken.

C. DEPARTMENT OF STATE CONTINUOUS CERTIFICATION AND ACCREDITATION PROCESS

The Department of State has developed a process for continuous Certification and Accreditation (see Figure 3) [18]. It is intended to conduct ongoing certification and accreditation that maps to the steps in the Risk Management Framework.

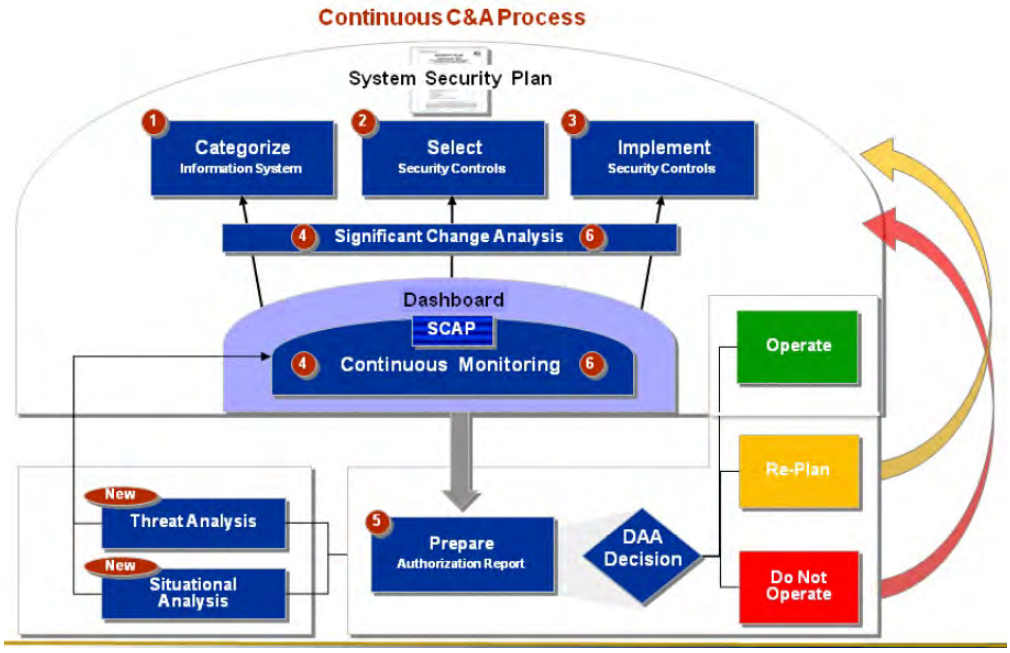


Figure 3. Department of State Continuous C&A Process

Step one is Categorize Information System. The information system is categorized and the System Security Plan is created in this step. The second step is to Select Security Controls. The System Security Plan and system categorization are used to select the security controls. System specific controls are also selected as appropriate. The selected controls are implemented in the third step: Implement Security Controls.

Significant Change Analysis is the fourth step in the process. Issues identified by the dashboard are evaluated to determine if further changes are needed. The fifth step, Continuous Monitoring, combines the fourth and sixth steps of the RMF which involve testing at two stages of the process: during certification and during monitoring. The Security Content Automation Protocol (SCAP) is used to test and communicate results to the dashboard and other steps in

the process. The final step is to Prepare Authorization Report. The dashboard provides a risk score used for the DAA Decision. The Threat Analysis looks at historical attacks to predict future events while the Situational Analysis looks at the current environment to enable effective actions. With the opportunity to catch errors early due to continuous monitoring testing, reaching *Do Not Operate* status should be extremely rare [18].

D. NAVY TRANSFORMATIONAL CERTIFICATION AND ACCREDITATION PROCESS

The Navy conducted a mapping between the DoDI 8500.2 and NIST SP 800–53 IA controls in order to combine the DIACAP and RMF processes into the Navy Transformational C&A Process. This process grew from the idea that "significant efficiencies can be gained through joint evaluations, and documentation, or overlapping security controls" [19].

This process consists of six events: Categorize Information System, Select Security Controls, Implement Security Controls, Assess Security Controls, Authorize Information System, and Monitor Security Controls. The tasks in each event are the combination of the DIACAP activities and RMF tasks.

E. OTHER IA PROCESSES

Other IA processes include the Connection Approval Process (CAP) [20] and Command Cyber Readiness Inspection (CCRI) [21]. The CJCSI 6211.02C, *Defense Information System Network (DISN): Policy, Responsibilities and Processes* [21], requires security controls to be in place in order for an IS to connect to the DISN and compliance inspections

to be conducted to ensure the continuing effectiveness of these controls. The CAP ensures the IS is secure and has an ATO before allowing it to connect to the DISN. The CCRI provides a “quick look” assessment of the network security configuration of an IS and its compliance with DoD IA and computer network defense (CND) policies. These processes could also be applicable to the concept presented in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. COMMON STRUCTURE

A. REDUNDANCY IN THE IA PROCESSES

From Chapter II it is evident that redundant activities are taking place. DISA has developed a mapping of the activities of the DIACAP to the steps in the RMF (see Appendix A). The steps of the aforementioned processes have been represented in Table 1. The common structure is added as the last row of the table to highlight the extent of the redundancy between the processes.

The concept proposed in this thesis is to turn this common structure into a continuous monitoring process and reduce redundancy and time. This process can be implemented in a tool that can incorporate process-specific documents and tasks to combine the various IA processes and reuse common data such as assessment results. In this manner, conducting the continuous monitoring process will in effect perform all processes it encompasses. Further redundancy can be reduced by synchronizing inspection and certification dates so that the results of one are still valid and applicable to the others.

Table 1. Steps of Various IA Processes

Process	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
DIACAP	Initiate and Plan IA C&A	Implement and Validate Assigned IA Controls	Make C&A Decision	Maintain ATO and Conduct Reviews	Decommission	
NIST RMF	Categorize Information System	Select Security Controls	Implement Security Controls	Assess Security Controls	Authorize Information System	Monitor Security Controls
DoS Continuous C&A Process	Categorize Information System	Select Security Controls	Implement Security Controls	Significant Change Analysis	Continuous Monitoring	Prepare Authorization Report
Navy C&A Transformational Process	Categorize Information System	Select Security Controls	Implement Security Controls	Assess Security Controls	Authorize Information System	Monitor Security Controls
Common Structure	Register or Update the System	Identify Security Controls	Implement Security Controls	Assess and Mitigate Security Controls	Determine and Accept Risk	Retire or Monitor the System

B. CONTINUOUS MONITORING PROCESS

Building upon the common structure discovered in Table 1, a continuous monitoring process has been developed. Figure 3 illustrates the process as a dynamic and flexible cycle with six activities.

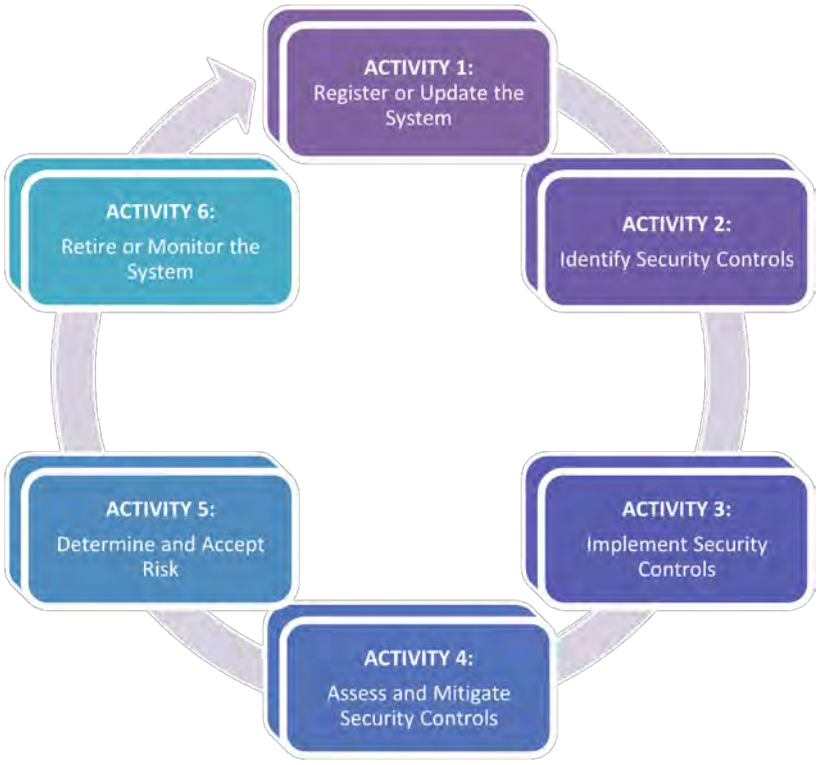


Figure 4. Continuous Monitoring Process

1. Register or Update the System

The first activity in this cycle is to register or update the information system. If the information system (IS) is new, registration will describe the system, the responsible entity and organization, the location, and other information that will be used to generate the required documents. The Mission Assurance Category (MAC) and Confidentiality Level (CL) of the system are determined

and used to establish the frequency of continuous monitoring reviews for the IS.

If the IS has already been registered, certified, and accredited, then this activity will be conducted when the IS or its information needs to be updated. This can be triggered if there are changes to the IS or its environment, key updates or patches for the IS, or a change in the MAC and/or CL of the IS. A scheduled review or re-certification will also launch this activity. Changes to the MAC or CL of the IS will change the applicable IA controls as well as the frequency with which reviews are conducted. Changes and updates need to be coordinated in order to ensure that these unintended results are considered and resolved. The updates occur in the first activity because the system will need to repeat the cycle to determine the security impact of the changes to the information system and its environment of operation. Repeating the cycle also ensures the IS remains compliant and secure.

2. Identify Security Controls

This activity uses the categorization information from activity one to assign the applicable base controls to the IS, as described in DODI 8500.2. Each of these controls will be identified as applicable, inherited, or not applicable, and all applicable controls will be determined to be either implemented or not implemented. The applicable but not implemented controls will be recorded in an Implementation Plan. All inherited controls require details of where the control is inherited from and a reason is required for all not applicable controls.

3. Implement Security Controls

This activity is the implementation of the relevant security controls in the Implementation Plan created in activity two. These controls are put into place and documented, as appropriate. The Implementation Results include the list of security controls implemented and a function description of the control implementation. The function description includes the planned inputs, expected behavior, and expected outputs. If a control failed to be implemented, the reason is entered in the Implementation Results accompanied by alternative methods of mitigation.

4. Assess and Mitigate Security Controls

All applicable controls should be implemented when activity four begins. This activity prepares an Assessment Plan that outlines the validation procedures of these controls, including automated scans (i.e. Gold Disk, Retina) and manual checklists. The Assessment Plan is then executed to determine the effectiveness of the controls, and ensure they have all been correctly implemented and work as intended.

The Assessment Results are used to prepare the POA&M. Remediation actions are conducted based on the POA&M. The POA&M is then updated based on the new Assessment Results. Remediation actions can be repeated as needed and if time permits. The issues, findings, and recommendations from the final Assessment Results and POA&M are documented in the Security Assessment Report.

5. Determine and Accept Risk

In activity five, the risk to organizational operations (including mission, functions, image or reputation), organizational assets, individuals, other organizations, or the nation is determined and documented in the Risk Assessment. The Security Assessment Report, from activity four, in conjunction with the Risk Assessment are submitted to the DAA or authorizing official to review for accreditation. If the risk is at an acceptable level, the DAA will accept it and issue the necessary Authorization Document, such as an Authorization to Operate (ATO) or Authorization to Connect (ATC), depending on the process taking place. A Monitor Strategy is developed with a plan to assess a selected subset of the technical, management, and operation security controls.

6. Retire or Monitor the System

At the end of the system’s life cycle, the system is decommissioned. The system registration information and system-related data are updated to reflect the system’s removal from active status.

If the system is not being retired, this activity will consist of monitoring activities until modifications or reviews take place. Ongoing security control assessment is conducted by executing the Monitor Strategy. This is followed by ongoing remediation actions and security status reporting. Activity one is initiated when updates to the system require changes to the system information or selected controls. A scheduled review, such as re-certification, can also initiate activity one. Ongoing risk determination and acceptance is conducted based on the

monitoring activities during periods when no major changes, updates, or reviews take place.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. CONTINUOUS MONITORING CONCEPT

The Continuous Monitoring process is the underlying structure of most IA processes. If a tool is created with this underlying structure, it can be used to conduct the various IA processes and house the process artifacts. This tool would be able to incorporate any process with the common structure identified in Chapter III. This can reduce the time required for each process and hold all information regarding an information system in one location. A prototype application was designed in order to assess the feasibility of this concept.

The application was developed using an MSI Wind Netbook with an Intel Atom processor and one GB of RAM. The operating system was Microsoft Windows XP Home Edition Version 2002 with Service Pack 3, although a full implementation of the proposed tool should be operating system independent. This proof-of-concept is designed as a three tiered web application that uses Java in order to connect the HTML pages with the PostgreSQL database [22]. The database management system used is pgAdmin Version 1.12.3 [23]. The database is configured with constant tables, constant views, and user-specific tables. Constant tables are initialized in the development of the application but not changed by any user. These tables include:

- MAC_Levels
  This table contains ‘MAC I’, ‘MAC II’, and ‘MAC III’ and is used as a reference in other tables to ensure

only these three values can be inserted as a MAC level.

- MAC_I_Controls
  This table contains a list of all the control numbers of controls for MAC I systems as defined in the DoDI 8500.2.

- MAC_II_Controls
  This table contains a list of all the control numbers of controls for MAC II systems as defined in the DoDI 8500.2.

- MAC_III_Controls
  This table contains a list of all the control numbers of controls for MAC III systems as defined in the DoDI 8500.2.

- Confidentiality_Levels
  This table contains ‘Public’, ‘Sensitive’, and ‘Classified’ and is used as a reference in other tables to ensure only these three values can be inserted as a Confidentiality level.

- Public_Controls
  This table contains a list of all the control numbers of controls for Public systems as defined in the DoDI 8500.2.

- Sensitive_Controls
  This table contains a list of all the control numbers of controls for Sensitive systems as defined in the DoDI 8500.2.

- Classified_Controls
  This table contains a list of all the control numbers of controls for Classified systems as defined in the DoDI 8500.2.

- Impact_Codes
  This table contains ‘High’, ‘Medium’, and ‘Low’ and is used as a reference in other tables to ensure only these three values can be inserted as an Impact Code.

- IA_Services
  This table contains ‘Availability’, ‘Confidentiality’, and ‘Integrity’ and is used as a reference in other tables to ensure only these three values can be inserted as an IA Service.

- Subject_Areas
  This table is used as a reference in other tables to ensure only valid values can be inserted as a Subject Area. The valid subject areas are ‘Continuity’, ‘Enclave and Computing Environment’, ‘Enclave Boundary Defense’, ‘Identification and Authentication’, ‘Personnel’, ‘Physical and Environmental’, ‘Security Design and Configuration’, and ‘Vulnerability and Incident Management’.

- IA_Controls
  This table contains all the IA Controls in the DoDI 8500.2. For each control, the table contains the control number, control name, IA service, impact code, subject area, and description. The IA service, impact code, and subject area reference the respective tables to reduce possible errors. The information for the

controls is taken from the DoDI 8500.2, and the impact codes for each control is taken from the DIACAP Knowledge Service.

- ScanTypes
  This table contains ‘GoldDisk’, ‘Retina’, and ‘Manual’ and is used as a reference in other tables to ensure only these three values can be inserted as a Scan Type. The values in this table will depend on the implementation and may include different types of scans.

The database contains user-specific tables that are initialized when a user gets an account and is changed at the will of the user. These tables include:

- UserName
  This table contains all the information systems that the user named UserName has registered with the application. Each user will have a unique username to sign into the application and this username will be the name of their table.

- UserScans
  For each user, this table contains all the scans conducted on the ISs registered to that user.

- IS_Controls
  In a fully functional implementation, each information system will have a table of controls associated with it. These will contain the base controls with any additions by the user. It will contain the control number, applicability (inherited / applicable / not applicable), comments on applicability (especially for

inherited or not applicable controls), implementation status (implemented / not implemented), comments on implementation status (i.e., why a control was not implemented), compliance status (inherited / not applicable / compliant / not compliant), comments on compliance status (i.e., why a control was not compliant). The last column will contain a variable that determines if the user can remove the control from the list. If the control is a base control, the user cannot remove it. If the user added the control, then the user can remove it at any time.

The database contains constant views that are created in the development of the application but not changed by any user. The views show the controls for the nine possible combinations of MAC and CL. Duplicate controls are removed and if there are two levels for the same control, the more secure level is chosen. The views are:

- MAC_I_Public
  This view shows only the controls for a MAC I Public system.
- MAC_I_Sensitive
  This view shows only the controls for a MAC I Sensitive system.
- MAC_I_Classified
  This view shows only the controls for a MAC I Classified system.
- MAC_II_Public
  This view shows only the controls for a MAC II Public system.

- MAC_II_Sensitive
  This view shows only the controls for a MAC II Sensitive system.
- MAC_II_Classified
  This view shows only the controls for a MAC II Classified system.
- MAC_III_Public
  This view shows only the controls for a MAC III Public system.
- MAC_III_Sensitive
  This view shows only the controls for a MAC III Sensitive system.
- MAC_III_Classified
  This view shows only the controls for a MAC III Classified system.

The environment the application uses is Apache Tomcat Version 6.0.28 [24]. It contains Catalina which is Tomcat's servlet container. Catalina implements Sun Microsystem's specifications for servlet and JavaServer Pages (JSP) [25]. The Java code is written and compiled in the Java SE Platform with JDK and JRE Version 1.6.0_26 [26]. The HTML pages [27] are generated from the Java classes in the apache folder when Catalina is initiated [28]. The pages are then viewed via Internet Explorer Version 8 [29].

The pages are organized to implement the activities of the Continuous Monitoring Process in Chapter III. Below is a description of the main pages implemented in this application. The Java code for these pages can be found in Appendix B.

A. REGISTER OR UPDATE THE SYSTEM

1. User Home Page

The sign-in and sign-out mechanisms are not applied in this application but assumed to be applied. After the user signs in, the user is directed to the UserHome page. This page displays all the registered information systems that the user has access to. The user selects a particular IS and is taken to the ISHome page to take action on that IS. The user can view all scans conducted for the listed information systems with the “View all Scans” link that will direct the user to the ViewScans page.

If the user has no registered information systems or wishes to register a new system, the “Register a New IS” link will take the user to the RegisterSystem page. The user can sign out of the Continuous Monitoring Program with the link at the top right. This “Sign out” link is on all the pages. See Figure 5 for a screenshot of the UserHome page of the proof-of-concept application.

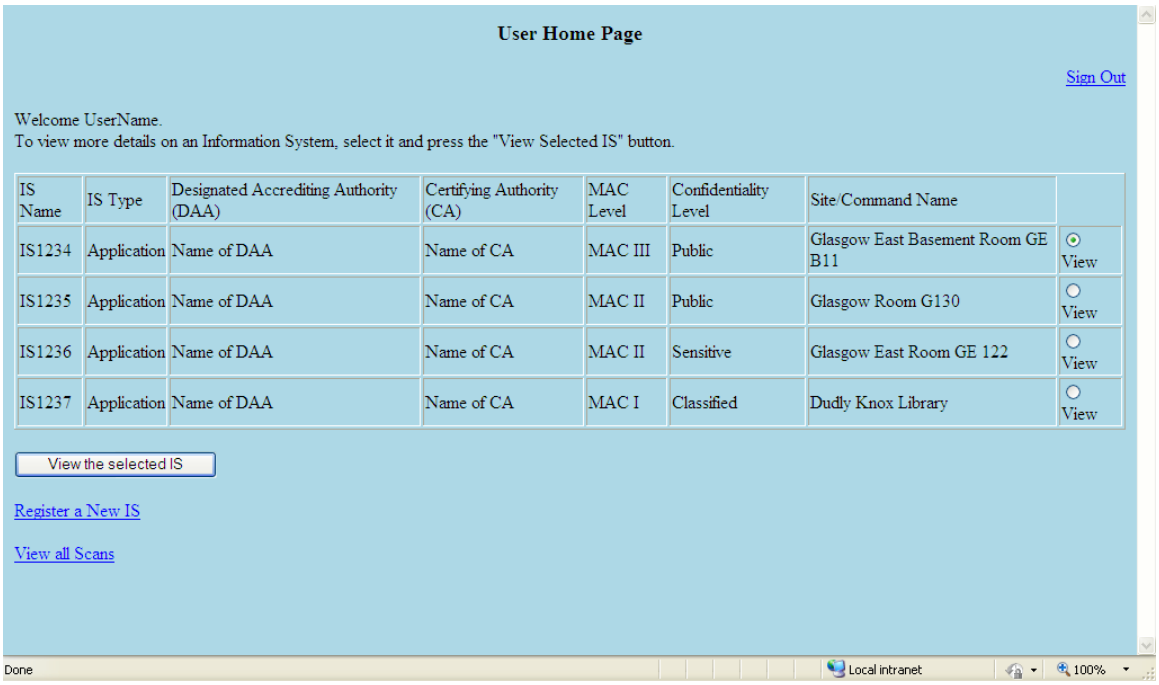


Figure 5. User Home Page

2. Information System Home

The ISHome page displays the details of the selected IS. From here the user can:

- Edit or update the details of the IS (EditSystem page)
- View the scans conducted on that particular IS (ISScans page)
- Upload a scan of that IS (UploadScan page)
- Retire the IS (RetireSystem page)
- Edit the IA Controls for the IS (ISControls page)
- View the IA Controls for the IS (ISControls page)
- Assess the IS (AssessSystem page)
- View/Accept the Risk of the IS (SystemRisk page)

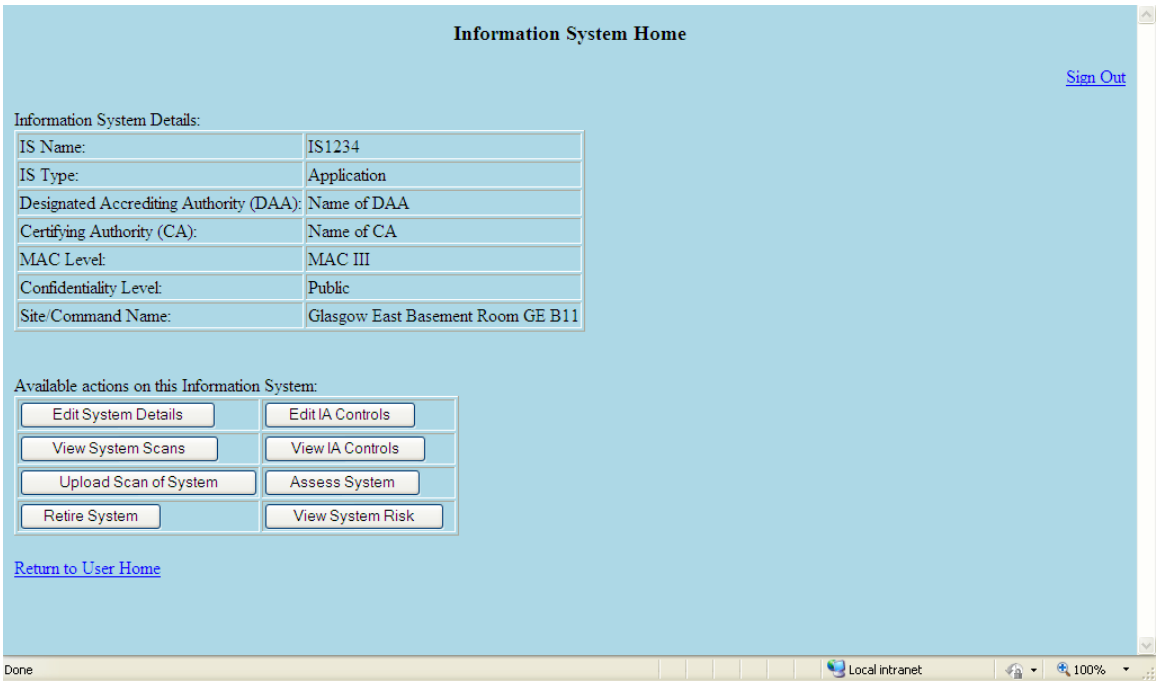


Figure 6. Information System Home Page

The user can also return to the UserHome page without making any changes to the selected IS. See Figure 6 for a screenshot of the ISHome page.

3. Register a System

The RegisterSystem page implements the Register part of Activity One of the Continuous Monitoring Process. It requests information from the user in order to register a new information system with the application. The information requested depends on the IA process or process step being carried out. See Figure 7 for a screenshot of the RegisterSystem page.

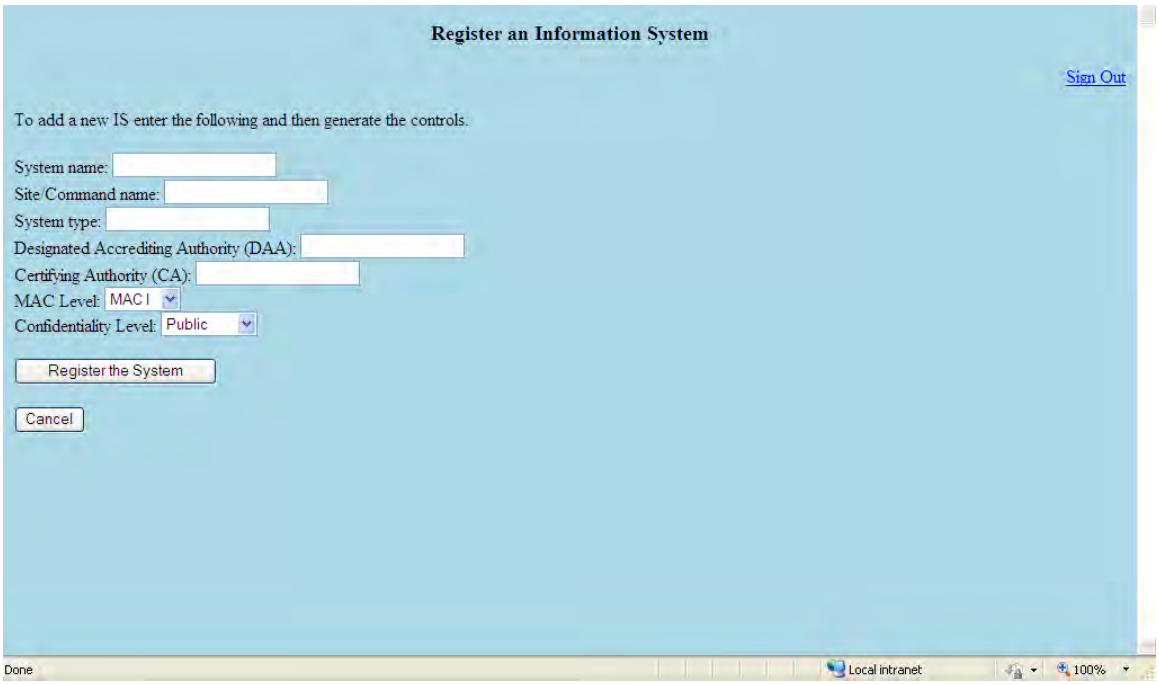


Figure 7. Register System Page

4. Edit a System

The Update part of Activity One of the Continuous Monitoring Process is implemented in the EditSystem page. This page displays the same information that was requested in the RegisterSystem page but allows the user to edit the details. The user can save the changes and return to ISHome to view the updated details of the IS. If the user does not want the changes to be saved, the "Cancel" button will return them to the ISHome page without making any changes.

B. IDENTIFY SECURITY CONTROLS

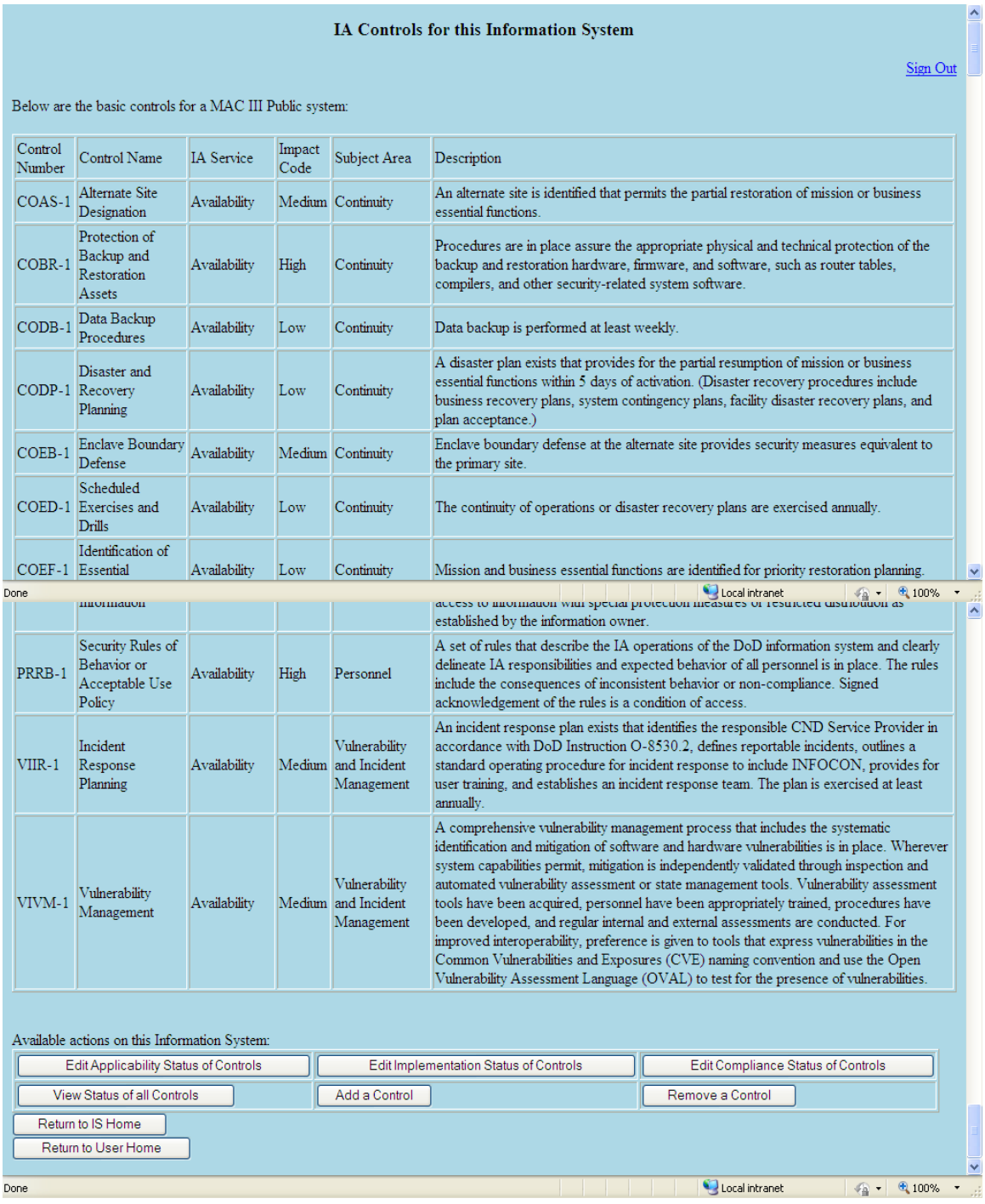
1. Identify and Select Controls

Activity Two of the Continuous Monitoring Process is conducted in the ISControls page. The application generates the base controls based on the MAC and CL of the IS. See

Figure 8 for a screenshot of the ISControls page. The user can then add or remove system-specific controls.

The user selects the “Edit Applicability Status of Controls” button and manually labels each control as Inherited, Applicable, or Not Applicable on the AppStatus page. A rationale is required for Inherited or Not Applicable controls. See Figure 9 for a screenshot of the AppStatus page.

The user then identifies which of the Applicable controls are already ‘Implemented’ and which are ‘Not Implemented’ on the ImpStatus page by selecting the “Edit Implementation Status of Controls” button. See Figure 10 for a screenshot of the ImpStatus page. The controls that are not implemented are compiled into an Implementation Plan that is to be conducted by the user.



IA Controls for this Information System

Sign Out

Below are the basic controls for a MAC III Public system:

Control Number	Control Name	IA Service	Impact Code	Subject Area	Description
COAS-1	Alternate Site Designation	Availability	Medium	Continuity	An alternate site is identified that permits the partial restoration of mission or business essential functions.
COBR-1	Protection of Backup and Restoration Assets	Availability	High	Continuity	Procedures are in place assure the appropriate physical and technical protection of the backup and restoration hardware, firmware, and software, such as router tables, compilers, and other security-related system software.
CODB-1	Data Backup Procedures	Availability	Low	Continuity	Data backup is performed at least weekly.
CODP-1	Disaster and Recovery Planning	Availability	Low	Continuity	A disaster plan exists that provides for the partial resumption of mission or business essential functions within 5 days of activation. (Disaster recovery procedures include business recovery plans, system contingency plans, facility disaster recovery plans, and plan acceptance.)
COEB-1	Enclave Boundary Defense	Availability	Medium	Continuity	Enclave boundary defense at the alternate site provides security measures equivalent to the primary site.
COED-1	Scheduled Exercises and Drills	Availability	Low	Continuity	The continuity of operations or disaster recovery plans are exercised annually.
COEF-1	Identification of Essential	Availability	Low	Continuity	Mission and business essential functions are identified for priority restoration planning.

Done Local intranet 100%

established by the information owner.

PRRB-1	Security Rules of Behavior or Acceptable Use Policy	Availability	High	Personnel	A set of rules that describe the IA operations of the DoD information system and clearly delineate IA responsibilities and expected behavior of all personnel is in place. The rules include the consequences of inconsistent behavior or non-compliance. Signed acknowledgement of the rules is a condition of access.
VIIR-1	Incident Response Planning	Availability	Medium	Vulnerability and Incident Management	An incident response plan exists that identifies the responsible CND Service Provider in accordance with DoD Instruction O-8530.2, defines reportable incidents, outlines a standard operating procedure for incident response to include INFOCON, provides for user training, and establishes an incident response team. The plan is exercised at least annually.
VIVM-1	Vulnerability Management	Availability	Medium	Vulnerability and Incident Management	A comprehensive vulnerability management process that includes the systematic identification and mitigation of software and hardware vulnerabilities is in place. Wherever system capabilities permit, mitigation is independently validated through inspection and automated vulnerability assessment or state management tools. Vulnerability assessment tools have been acquired, personnel have been appropriately trained, procedures have been developed, and regular internal and external assessments are conducted. For improved interoperability, preference is given to tools that express vulnerabilities in the Common Vulnerabilities and Exposures (CVE) naming convention and use the Open Vulnerability Assessment Language (OVAL) to test for the presence of vulnerabilities.

Available actions on this Information System:

Edit Applicability Status of Controls | Edit Implementation Status of Controls | Edit Compliance Status of Controls

View Status of all Controls | Add a Control | Remove a Control

Return to IS Home

Return to User Home

Done Local intranet 100%

Figure 8. IS Controls Page

Figure 9. Applicability Status of Controls Page

Figure 10. Implementation Status of Controls Page

C. IMPLEMENT SECURITY CONTROLS

The user takes the Implementation Plan generated in the ISControls page and implements the security controls. The Implementation Results are used to manually update the ImpStatus page, changing Not Implemented to Implemented on appropriate controls. Activity Three does not have its own page but makes use of the ImpStatus page (See Figure 10).

D. ASSESS AND MITIGATE SECURITY CONTROLS

1. Viewing the Scans

The ViewScans page displays all the scans conducted on the user's registered information systems. The user can return to UserHome after viewing the existing scans or upload a new scan. The "Upload a new Scan" button directs the user to the UploadScan page. A screenshot of the ViewScans page is in Figure 11. If the user wants to view scans for a particular IS, the "View System Scans" button on the ISHome page will take the user to the ISScans page. This page looks like the ViewScans page, but only displays the scans for the selected IS.

Figure 11. View Scans Page

2. Upload a Scan

The UploadScan page requests information about the scan being uploaded and adds it to the table of scans. The capability of conducting a scan through the application has not been implemented. However, including the scan ability in the application can allow for automatic, scheduled scans to take place. These potential additions are discussed further in Chapter V.

3. Assess the System

The user generates an Assessment Plan that outlines the validation procedures and then implements the plan. This may include conducting automated scans, manual checks, or other actions. The AssessSystem page generates a Plan of Action and Milestone (POA&M) document from the results of the most recent scan of the IS. This requires at least one scan of the system to be uploaded to the application. The application generates a table with the selected controls for the IS and includes the Inherited and Not Applicable labels and comments. For controls previously labeled as Applicable, the user must now enter if the control is Compliant or Not Compliant based on the results of the scan. Figure 12 illustrates the CompStatus page that allows for this labeling.

Compliance Status of Controls

Sign Out

Label each control as Compliant or Not Compliant.

control_number	control_name	impact_code	applicability	implementation	compliance
COAS-1	Alternate Site Designation	Medium	Inherited	Inherited	Inherited
COBR-1	Protection of Backup and Restoration Assets	High	Not Applicable	Not Applicable	Not Applicable
CODB-1	Data Backup Procedures	Low	Applicable	Not Implemented	Compliant
CODP-1	Disaster and Recovery Planning	Low	Applicable	Implemented	Compliant
COEB-1	Enclave Boundary Defense	Medium	Applicable	Implemented	Compliant
COED-1	Scheduled Exercises and Drills	Low	Applicable	Implemented	Compliant
COEF-1	Identification of Essential Functions	Low	Applicable	Implemented	Compliant
COMS-1	Maintenance Support	Low	Applicable	Implemented	Compliant
PEVR-1	Voltage Regulators	High	Applicable	Implemented	Compliant
PRMP-1	Maintenance Personnel	High	Applicable	Implemented	Compliant
PRNK-1	Access to Need-to-Know Information	High	Applicable	Implemented	Compliant
PRRB-1	Security Rules of Behavior or Acceptable Use Policy	High	Applicable	Implemented	Compliant
VIIR-1	Incident Response Planning	Medium	Applicable	Implemented	Compliant
VIVM-1	Vulnerability Management	Medium	Applicable	Implemented	Compliant

Save changes

Return to IS Controls

Return to IS Home

Return to User Home

Done Local intranet 100%

Figure 12. Compliance Status of Controls Page

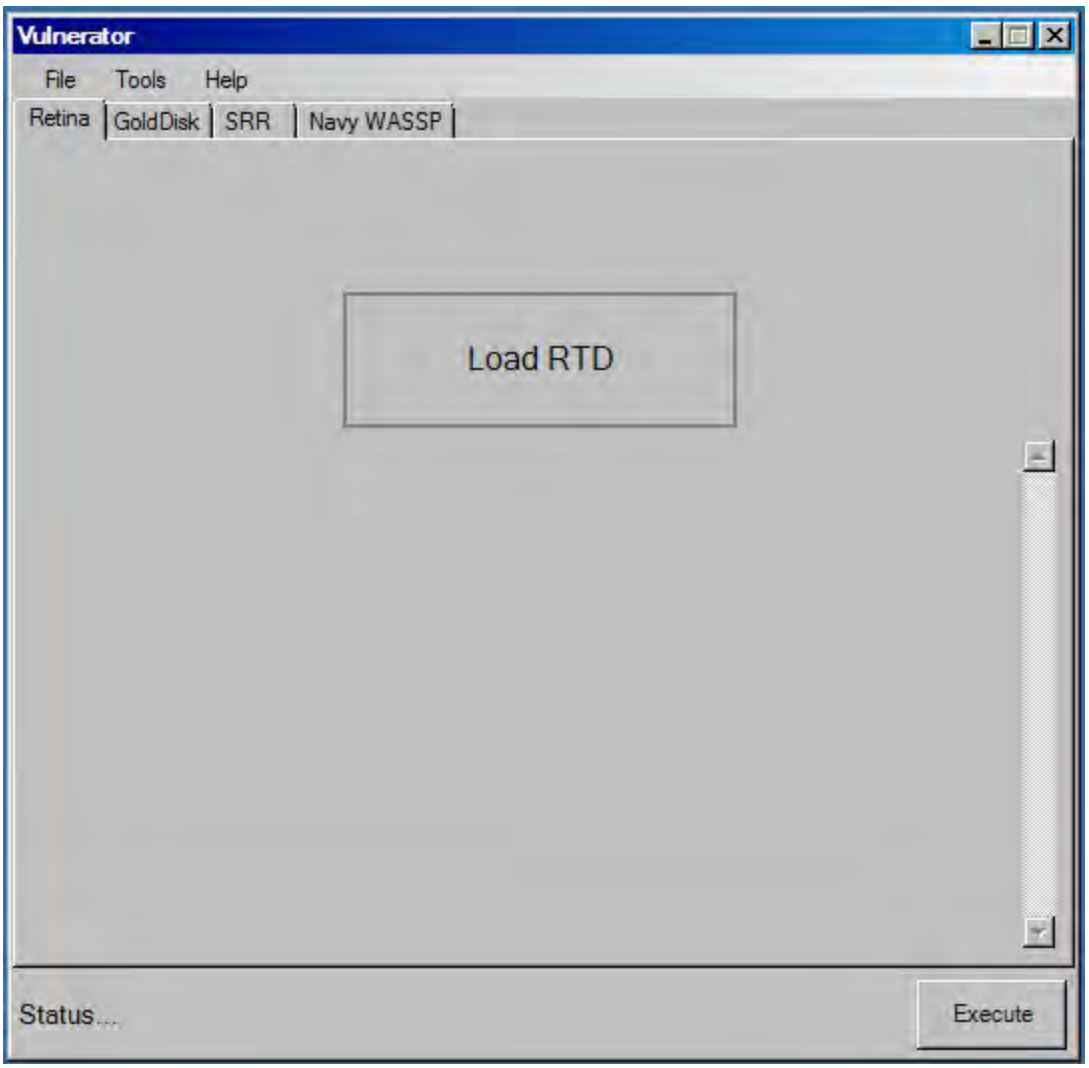


Figure 13. Vulnerator Program

There exists a parsing tool called the Vulnerator which parses Gold Disk, Retina, SRR, and Navy WASSP results into user-friendly Excel documents. See Figure 13 for an illustration of the Vulnerator Program. The user clicks on the center button “Load RTD” and browses for the RTD scan to upload. RTD is the file extension of Retina scans. The other tabs ask for the respective scan result files to be uploaded. Once uploaded, the user selects the “Execute” button to begin parsing. The outcome is an Excel

representation of the results of the uploaded scan. The user could use the Vulnerator to help with the input of results though this method is still prone to human error. The full implementation could include a tool such as the Vulnerator in order to automate this section (more in Chapter V). This assessment of the compliance of the selected IA controls is the first part of Activity Four. It determines the effectiveness of the controls and ensures they are implemented and working as intended.

4. Mitigate Controls

The second part of Activity Four is to mitigate non compliant controls based on the POA&M. There is no page for this in the proof-of-concept application because the user must do this outside of the application.

E. DETERMINE AND ACCEPT RISK

1. Accept the Risk

Both the Determine and Accept parts of Activity Six are conducted through the SystemRisk page. The Overall System Risk Score is calculated using the most recent POA&M results and the impact codes for each outstanding control. The authorizing official reviews the Security Assessment Report and then decides to either accept or reject the risk. This decision is based on what this score and information system mean with respect to organizational operations, organizational assets, individuals, other organizations, and the nation. To ensure this decision is not falsely entered into the system, the full implementation of this proof-of-concept should use digital signatures. Depending on the specific IA process being

implemented through this application, other documents may be generated. For example, the authorizing official will issue an Authorization Document, such as an ATO, at this stage.

The impact code for each control number in the DoDI 8500.2 can be found at the DIACAP Knowledge Service and was hardcoded into a table in the database. [30] The application determines the number of high, medium, and low impact controls are non compliant. Depending on the IA process, the application may use this information in a process-specific formula or method of calculating the Overall System Risk Score. If the IA processes have different formulas or methods, a standard for the formula should be established. For the proof-of-concept application, the formula below was implemented in order to demonstrate the step.

*Overall Risk Score (%) = HighNum*10 + MediumNum*5 + LowNum*

This formula claims that a non compliant high impact control is a 10% risk, a non compliant medium impact control is 5% risk, and a non compliant low impact control is 1% risk. See Figure 14 for a screenshot of the SystemRisk page.

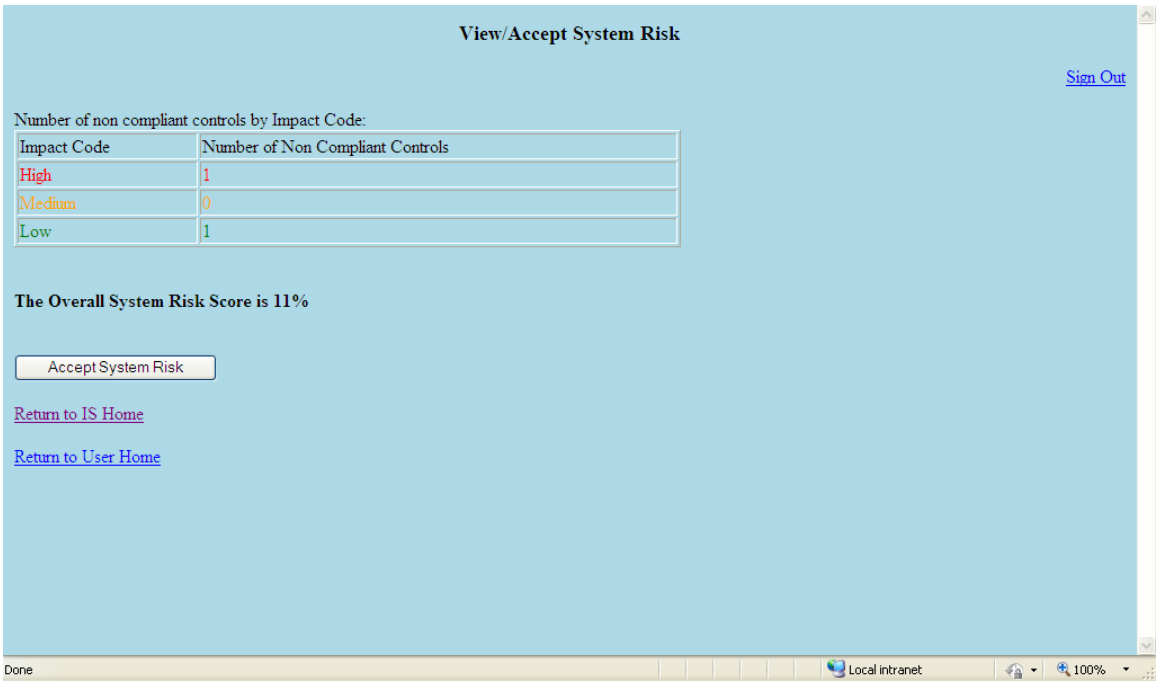


Figure 14. System Risk Page

F. RETIRE OR MONITOR THE SYSTEM

1. Retire the System

The RetireSystem page displays the system's information and asks if the user is sure about retiring the system. The user can "Cancel" and return to the ISHome page, or "Retire the IS" and return to the UserHome page. If user retires the system, all scans of that system, its information, and any documents associated with it are removed from the application. This page implements the Retire part of Activity Six. See Figure 15 for a screenshot of the RetireSystem page.

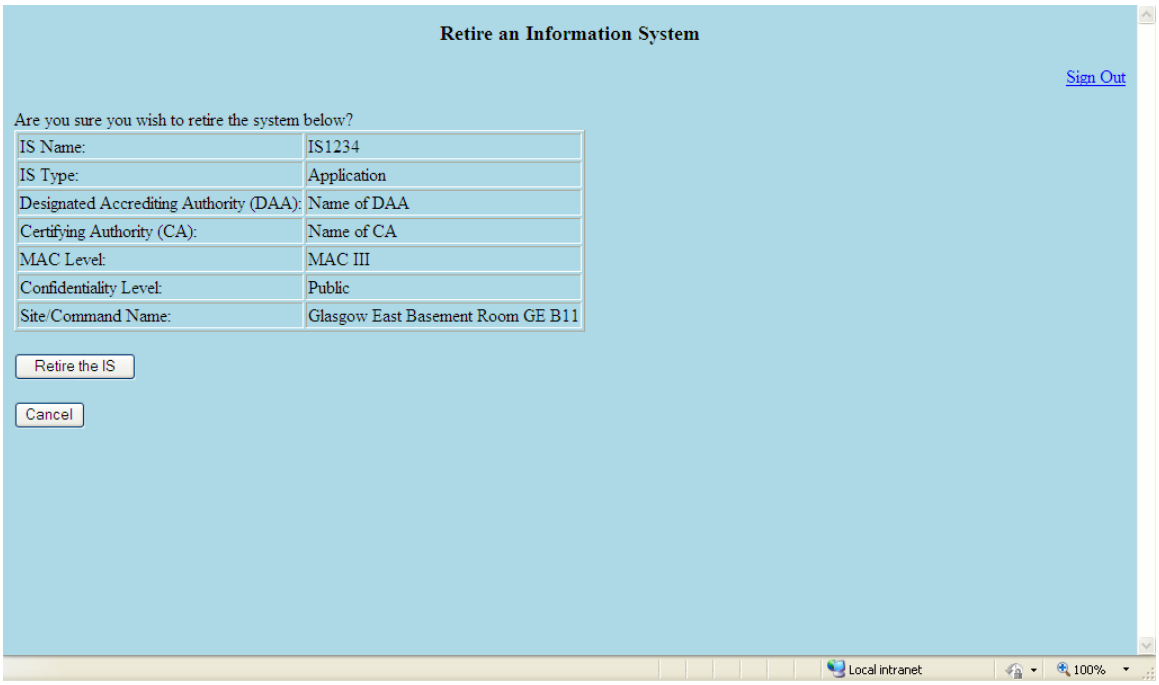


Figure 15. Retire System Page

2. Monitor the System

There is no page for the Monitor part of Activity Six. To monitor the system, the user first updates the system through the EditSystem page as appropriate. The user then conducts a scan on the system, uploads the scan results through the UploadScan page, and assesses the system based on the scan through the AssessSystem page. This is repeated at a frequency commensurate with the level of the IS.

V. CONCLUSION

A. RESULTS

The proof-of-concept demonstrates the similar structure of the four IA processes discussed in Chapter II. If the details of each process are removed, such as the tools used and documents generated, the construction of the process will look similar to this:

- Register or Update the System
- Identify Security Controls
- Implement Security Controls
- Assess and Mitigate Security Controls
- Determine and Accept Risk
- Retire or Monitor the System

The proof-of-concept takes the Continuous Monitoring Process of Chapter III and presents it as an application. A user signs in and has access to view or edit all their registered ISs. The application can be modified to turn the information on a particular IS into a process-specific document based on a template. In this manner, a larger tool can be developed that incorporates all relevant IA processes. Information can be shared and scans can be reused in order to avoid redundancy between submitting an IS through more than one process. If the processes are conducted around the same time, the information and scans will still be valid. This will reduce time as the user will not have to conduct another scan on the IS.

B. AUTOMATION

The proof-of-concept tool revealed that some key parts require a tedious amount of work by the user. This increases the probability of introducing human errors. Automation of those sections would save time and reduce human error. Human intervention is still necessary in some areas. For instance, Activity One should not be automated. Adding or editing an information system should be conducted by a system administrator or by the IS owner. There should be audit logs to provide accountability and ensure the integrity of the data.

Activity Two can be semi-automated. If a tool can be developed to identify the hardware and software on an IS, the system-specific controls can be automatically added to the list of applicable controls. The software can also be checked against the DoD authorized versions. Automation in Activity Three is desirable but complicated. The Gold Disk tool has the capability for automated implementation of some controls. However, the implementation of one control may break another. The method of implementation for each control and its effect on the system should be considered if developing an automated implementation tool. The tool could automatically update the database by labeling the controls as compliant or not compliant. The inherited and not applicable controls can carry forward from Activity Two.

Current tools relevant to Activity Four are mostly vulnerability scanners such as Retina. A new tool of that type could be developed to assess the effectiveness of more security controls. However, the more pressing need is to

develop a parsing tool to automate the parsing of scan results. Such a tool exists for Retina, Gold Disk, SRR, and Navy WASSP results called the Vulnerator (see Figure 13). If a new vulnerability scanner is employed, a new tool must be developed to parse the results. The proof-of-concept tool only considers an upload of a scan, however incorporating the vulnerability scanner and parsing tool could further automate this activity. An information system could then be scanned directly from the Continuous Monitoring tool, either by signing in to the application while on the IS to be scanned, or by conducting the scan remotely. The assessments of the scans are used to mitigate the outstanding controls. The automation of this process presents the same apprehension as the automation of implementing controls. The remediation of some controls may not be desirable or may require an examination of possible effects prior to mitigation.

Activity Five can automate the determination of risk by calculating the Overall System Risk Score based on the impact codes of outstanding IA controls. Acceptance of that risk requires a human to decide based on information not available to the tool, such as the situation or environment. Decommissioning a system in Activity Six is also an action that should not be automated. Monitoring the system can be automated by scheduling automated scans to be run at a frequency commensurate with the MAC and CL of the IS.

C. FUTURE WORK

Future work includes research into developing the aforementioned automated tools and conducting a security

analysis on the Vulnerator tool. Once the automation is accomplished, a fully functional and automated tool can be developed and the various IA processes can be added to it. To incorporate a particular IA process, the process-specific information needs to be requested from the user and the output documents need to be generated at the respective stages of the process.

When the relevant processes are incorporated, security mechanisms should be added to address vulnerabilities associated with signing in, access control, Java, PostgreSQL, HTML, and other relevant aspects of the tool. A security assessment of the Continuous Monitoring automated tool will ensure that these considerations are addressed and that the application complies with DoD STIGs and other requirements. It can also be evaluated for effective continuous monitoring by ensuring it contains the elements outlined in NIST SP 800–137, *Information Security Continuous Monitoring for Federal Information Systems and Organizations.* [31]

D. LONG-TERM CHALLENGES

A considerable amount of work needs to be performed before this concept can be fully functional and effective. During that time new processes may be created. The advantage of the proposed Continuous Monitoring automated tool is that it will still be relevant despite changing IA processes or new vulnerabilities. New vulnerabilities require all susceptible systems to be updated and relevant mitigations to be applied. Thus, this is not a shortcoming of the tool. New IA processes will most likely have the structure outlined in this thesis and therefore still be

compatible with the proposed tool. Therefore, a fully functional and automated version of the Continuous Monitoring tool is feasible. It would not only be able to implement any IA process with this structure, but also reduce redundancy, error, and time. For that reason, more time should be spent on developing automation tools instead of on creating new IA processes.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. MAPPING OF DIACAP AND RMF ACTIVITIES

Cross-walk between NIST SP 800–37 and DoDI 8510.01 Activities created by DISA FSO and taken from the DIACAP Knowledge Service (KS) [10].

Mapping of Subtasks/Documents with Comments	
DODI 8510.01 DIACAP Activities	**NIST SP 800–37 Steps**
1.a **Register System with DoD Component IA Program** • Develop System Identification Profile	1–2 **Information System Description** • Document system identification section of Security Plan 1–3 **Information System Registration** • Registration is managed by the organization
1.b **Assign IA Controls** • Document IA controls in the DIACAP Implementation Plan	1–1 **Security Categorization** • RMF applied and categorization of individual information system accomplished as an organization-wide activity 2–1 **Common Control Identification** • Identification of security controls is a separate step with the common controls documented in the Security Plan, Note: Inherited controls are listed as inherited on the DIP (DIP). 2–2 **Security Control Selection** • Baseline and supplemental. Document in *Security Plan.*
1.c **Assemble DIACAP Team**	
1.d **Initiate DIACAP Implementation Plan** • Maps to 800–37, Implement Step	
	2–3 **Monitoring Strategy** • This step is accomplished in DIACAP activity 4.a
	2–4 **Security Plan Approval**

	• 800–37 requires authorization official approval, DIACAP does not
2a. **Execute DIACAP Implementation Plan (DIP)** • Execute/update the DIP/use KS to implement	3–1: **Security Control Implementation** • Implements controls listed in *Security Plan* 3–2: **Security Control Documentation** • Document implementation (planned inputs, expected behavior, and expected outputs) in *Security Plan*. **Note:** Implementation procedures are standardized in KS
2b. **Conduct Validation Activities** • Execute IAW the KS. Note: DIACAP is mute on developing an assessment plan around, or in addition to the validation procedures in the KS.	4–1: **Assessment Preparation** • Develop a Security Assessment Plan and get it approved 4–2: **Security Control Assessment** • Execute the assessment
2c. **Prepare POA&M** 2d. **Compile Validation Results in DIACAP Scorecard** • Prepare POA&M and populate the Scorecard • Map to 800–37 Step 5 Authorize IS	**Note:** DIACAP is mute on remediation actions prior to submitting the Scorecard and POA&M to the DAA. Logic would say that if an IA control deficiency could be remediated immediately then it would. If additional resources must be applied for remediation, then it possibly may be deferred and remediated later.
3a. **Make Certification Determination** • Severity codes assigned to POA&M. CA signs Scorecard and forwards either an Executive or Comprehensive Package to DAA	5–3: **Risk Determination**
3b. **Issue Accreditation Decision** • DAA signs and assigns authorization termination	5–4: **Risk Acceptance** • The authorizing official generates an *Authorization Decision Document*

date on Scorecard	
	Note: 4–1: Assessment Preparation • Takes place in DIACAP Activity 2c. 4–3: Security Assessment Report • Security Authorization Package contains the Security Plan, Security Assessment Report, & POA&M. Takes place in DIACAP Activity 2d.
4a. **Maintain Situational Awareness**• Develops System Monitoring Program	6–1: **IS and Environment Changes**6–2: **Ongoing Security Control Assessments**6–3: **Ongoing Remediation Actions** • Generates changes in the Security Assessment Report
	Note: NIST Step 2–3: Refers to developing continuous monitoring program and documenting in the Security Plan
4b. **Maintain IA Posture** 4c. **Conduct Reviews** 4d. **Initiate Re-accreditation** • Generates changes to DIACAP Package and reports to DAA	6–3: **Ongoing Remediation Actions** 6–4: **Key Updates** 6–5: **Security Status Reporting** 6–6: **Ongoing Risk Determination and Acceptance**
	Note: Step 6–7 occurs during DIACAP Activity 5a.
5a. **Retire System**	6–7: **IS Removal & Decommissioning**

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. PROOF-OF-CONCEPT CODE

The java code for the various pages of the proof-of-concept application:

A. USER HOME

http://localhost/examples/servlets/servlet/UserHome

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class UserHome extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = “<!DOCTYPE HTML PUBLIC \”-//W3C//DTD
HTML 4.0 “ + “Transitional//EN\”>\n”;

        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();
        out.println(“<html>“);
        out.println(“<head>“);
        out.println(“<title>Continuous Monitoring Program: User
Home</title>“);
        out.println(“</head>“);
        out.println(“<body style=\”background-
color:lightblue\”>“);
           out.println(“<h3 align=\”center\”>“);
        out.println(“User Home Page</h3>“);
           out.println(“<p align=\”right\”>“);
           out.println(“<a
href=\”http://localhost/examples/servlets/servlet/LogOutPage\”>Si
gn Out</a>“);
           out.println(“</p>“);
        out.println(“Welcome UserName.<br>“);
           out.println(“To view more details on an Information
System, select it and press the \”View Selected IS\”
button.<br><br>“);

           String tempName = ““;
```

```
          String RowName = “”;

          String query1 = “update username set inuse=‘n’ where
inuse=‘y’”;
          if (query1 != null )
          {
          try
               {
                    Class.forName(“org.postgresql.Driver”);
                    System.out.println(“Driver loaded”);
                    String
url=“jdbc:postgresql://localhost/IAcontrols”;
                    String user = “postgres”;
                    String pwd = “postgres”;
                    Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                    System.out.println(“Database Connect ok”);

                    Statement
query_stmt1=DB_mobile_conn.createStatement();
                    ResultSet
query_rs1=query_stmt1.executeQuery(query1);
                    query_rs1.close();
                    query_stmt1.close();
               } catch (Exception exp)
               {
                    System.out.println(“Query exception = “
+exp+ “\n”);
           }
          }

          String query2 = “update username set inuse=‘y’ where
isname=‘“;
          String query = “select * from username”;
          if (query != null )
          {
               try
               {
                    Class.forName(“org.postgresql.Driver”);
                    System.out.println(“Driver loaded”);
                    String
url=“jdbc:postgresql://localhost/IAcontrols”;
                    String user = “postgres”;
                    String pwd = “postgres”;
                    Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                    System.out.println(“Database Connect ok”);
                    Statement
query_stmt=DB_mobile_conn.createStatement();
```

```
                        ResultSet
query_rs=query_stmt.executeQuery(query);
                        ResultSetMetaData rsmd =
query_rs.getMetaData();
                        int queryColCount = rsmd.getColumnCount();

                        out.println("<form action=\"UserHome\"
method=POST>");
                        out.println("<table border=\"1\">");

                        String colName="";
                        colName = colName + "<td>IS Name</td>" +
"<td>IS Type</td>" + "<td>Designated Accrediting Authority
(DAA)</td>";
                        colName = colName + "<td>Certifying
Authority (CA)</td>" + "<td>MAC Level</td>" +
"<td>Confidentiality Level</td>";
                        colName = colName + "<td>Site/Command
Name</td>";
                        out.println("<tr>"+colName + "</tr>" );

                        while (query_rs.next())
                        {
                              sc.log("Column Returned");
                              String row = "";
                              for (int col=1; col <
queryColCount;col++)
                              {
                                    if (col == 1) tempName =
query_rs.getString(col);
                                    row = row + "<td>"+
query_rs.getString(col)+"</td>" ;
                              }
                              out.println("<tr>"+row + "<td>");
                              out.println("<input type=\"radio\"
name=\"RowName\" value=\"" + tempName + "\" /> View<br />");
                              out.println("</td></tr>");
                        }
                        out.println("</table>");
                        query2 = query2 + RowName + "'";
                        out.println("<br>");
                        out.println("<input type=submit value=" +
"\"View the selected IS\"" + ">");
                        out.println("</form>");

                        query_rs.close();
                        query_stmt.close();

                  } catch (Exception exp)
                  {
```

```
                    System.out.println("Query exception = "
+exp+ "\n");
             }
            }

            if (query2 != null )
            {
            try
                  {
                        Class.forName("org.postgresql.Driver");
                        System.out.println("Driver loaded");
                        String
url="jdbc:postgresql://localhost/IAcontrols";
                        String user = "postgres";
                        String pwd = "postgres";
                        Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                        System.out.println("Database Connect ok");

                        Statement
query_stmt2=DB_mobile_conn.createStatement();
                        ResultSet
query_rs2=query_stmt2.executeQuery(query2);
                        out.println("QUERY2: " + query2 + "<br>");

                        query_rs2.close();
                        query_stmt2.close();

                  } catch (Exception exp)
                  {
                        System.out.println("Query exception = "
+exp+ "\n");
             }
            }

            out.println("<P>");
            out.println("<p align=\"left\">");
            out.println("<a
href=\"http://localhost/examples/servlets/servlet/RegisterSystem\
">Register a New IS</a>");
            out.println("</p>");
            out.println("<p align=\"left\">");
            out.println("<a
href=\"http://localhost/examples/servlets/servlet/ViewScans\">Vie
w all Scans</a>");
            out.println("</p>");
          out.println("</body>");
          out.println("</html>");
      }
```

```
         public void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

B. INFORMATION SYSTEM HOME

http://localhost/examples/servlets/servlet/ISHome

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ISHome extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = “<!DOCTYPE HTML PUBLIC \”-//W3C//DTD
HTML 4.0 “ + “Transitional//EN\”>\n”;

        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();
        out.println(“<html>“);
        out.println(“<head>“);
        out.println(“<title>Continuous Monitoring Program: IS
Home</title>“);
        out.println(“</head>“);
        out.println(“<body style=\”background-
color:lightblue\”>“);
        out.println(“<h3 align=\”center\”>Information System
Home</h3>“);
           out.println(“<p align=\”right\”>“);
           out.println(“<a
href=\”http://localhost/examples/servlets/servlet/LogOutPage\”>Si
gn Out</a>“);
           out.println(“</p>“);
        out.println(“Information System Details: <br>“);

           String query = “select * from username where inuse
like ‘y’”;
```

```
          if (query != null)
          {
         try
               {
                     Class.forName(“org.postgresql.Driver”);
                     String
url=“jdbc:postgresql://localhost/IAcontrols”;
                     String user = “postgres”;
                     String pwd = “postgres”;
                     Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                     Statement
query_stmt=DB_mobile_conn.createStatement();
                     ResultSet
query_rs=query_stmt.executeQuery(query);
                     ResultSetMetaData rsmd =
query_rs.getMetaData();
                     int queryColCount = rsmd.getColumnCount();
                     out.println(“<table border=\”1\”>“);

                     while (query_rs.next())
                     {
                           sc.log(“Column Returned”);
                           String row = “”;
                           int col=1;
                           row = “<td>IS Name: </td>“ + “<td>“+
query_rs.getString(col)+”</td>“;
                           out.println(“<tr>“+row + “</tr>“);
                           col++;
                           row = “<td>IS Type: </td>“ + “<td>“+
query_rs.getString(col)+”</td>“;
                           out.println(“<tr>“+row + “</tr>“);
                           col++;
                           row = “<td>Designated Accrediting
Authority (DAA):</td>“ + “<td>“+ query_rs.getString(col)+”</td>“;
                           out.println(“<tr>“+row + “</tr>“);
                           col++;
                           row = “<td>Certifying Authority (CA):
</td>“ + “<td>“+ query_rs.getString(col)+”</td>“;
                           out.println(“<tr>“+row + “</tr>“);
                           col++;
                           row = “<td>MAC Level: </td>“ + “<td>“+
query_rs.getString(col)+”</td>“;
                           out.println(“<tr>“+row + “</tr>“);
                           col++;
                           row = “<td>Confidentiality Level:
</td>“ + “<td>“+ query_rs.getString(col)+”</td>“;
                           out.println(“<tr>“+row + “</tr>“);
                           col++;
```

```
                              row = “<td>Site/Command Name: </td>“ +
“<td>“+ query_rs.getString(col)+”</td>“;
                              out.println(“<tr>“+row + “</tr>“);
                         }
                         out.println(“</table>“);

                         query_rs.close();
                         query_stmt.close();

                    } catch (Exception exp)
                    {
                         out.println(“Query exception = “ +exp+
“\n”);
                         System.out.println(“Exception = “ +exp);
                     }
           } else
              {
               out.println(“No Query, Please enter “);
           }
           //out.println(“<P>“);
              out.println(“<br><br>“);
              out.println(“Available actions on this Information
System: <br>“);

              out.println(“<table border=\”1\” width=\”40%\”>“);
              out.println(“<tr>“);
              out.print(“<td width=\”20%\”>“);
              out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/EditSystem’\”>“);
              out.println(“Edit System Details</button></td>“);
              out.print(“<td width=\”20%\”>“);
              out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/EditControls’\”>“);
              out.println(“Edit IA Controls</button></td>“);
              out.println(“</tr>“);
              out.println(“<tr>“);
              out.print(“<td width=\”20%\”>“);
              out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/ISScans’\”>“);
              out.println(“View System Scans</button></td>“);
              out.print(“<td width=\”20%\”>“);
              out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/ISControls’\”>“);
              out.println(“View IA Controls</button></td>“);
              out.println(“</tr>“);
              out.println(“<tr>“);
```

```
            out.print(“<td width=\”20%\”>“);
            out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/UploadScan’\”>“);
            out.println(“Upload Scan of System</button></td>“);
            out.print(“<td width=\”20%\”>“);
            out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/AssessSystem’\”>“);
            out.println(“Assess System</button></td>“);
            out.println(“</tr>“);
            out.println(“<tr>“);
            out.print(“<td width=\”20%\”>“);
            out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/RetireSystem’\”>“);
            out.println(“Retire System</button></td>“);
            out.print(“<td width=\”20%\”>“);
            out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/SystemRisk’\”>“);
            out.println(“View System Risk</button></td>“);
            out.println(“</tr>“);
            out.println(“</table>“);

            out.println(“<p align=\”left\”>“);
            out.println(“<a
href=\”http://localhost/examples/servlets/servlet/UserHome\”>Retu
rn to User Home</a>“);
            out.println(“</p>“);

            out.println(“</body>“);
        out.println(“</html>“);
    }

    public void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

C. REGISTER SYSTEM

http://localhost/examples/servlets/servlet/RegisterSystem

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class RegisterSystem extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD
HTML 4.0 " + "Transitional//EN\">\n";

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Register an Information
System</title>");
        out.println("</head>");
        out.println("<body style=\"background-
color:lightblue\">");
           out.print("<h3 align=\"center\">");
        out.println("Register an Information System</h3>");
           out.println("<p align=\"right\">");
           out.println("<a
href=\"http://localhost/examples/servlets/servlet/LogOutPage\">Si
gn Out</a>");
           out.println("</p>");
        out.println("To add a new IS enter the following and then
generate the controls.<br>");
           String query=null;

           String isname = request.getParameter("isname");
           String istype = request.getParameter("istype");
           String daa = request.getParameter("daa");
           String ca = request.getParameter("ca");
           String mac_level = request.getParameter("mac_level");
           String location = request.getParameter("location");
           String conf_level =
request.getParameter("conf_level");

        out.println("<P>");
```

```
          out.println("<form action=\"RegisterSystem#runquery\"
method=POST>");
        out.println("System name:");
        out.println("<input type=text size=20 name=isname>");
        out.println("<br>");
           out.println("Site/Command name:");
        out.println("<input type=text size=20 name=location>");
        out.println("<br>");
           out.println("System type:");
        out.println("<input type=text size=20 name=istype>");
        out.println("<br>");
           out.println("Designated Accrediting Authority
(DAA):");
        out.println("<input type=text size=20 name=daa>");
           out.println("<br>");
           out.println("Certifying Authority (CA):");
        out.println("<input type=text size=20 name=ca>");
           out.println("<br>");
           out.println("MAC Level:");
           out.println("<select name=mac_level>");
           out.println("<option values="+"MAC I"+">MAC
I</option>");
           out.println("<option values="+"MAC II"+">MAC
II</option>");
           out.println("<option values="+"MAC III"+">MAC
III</option>");
           out.println("</select>");
           out.println("<br>");
        out.println("Confidentiality Level:");
           out.println("<select name=conf_level>");
           out.println("<option
values="+"Public"+">Public</option>");
           out.println("<option
values="+"Sensitive"+">Sensitive</option>");
           out.println("<option
values="+"Classified"+">Classified</option>");
           out.println("</select>");
           out.println("<br>");
     int queryComplete=0;
           query = "insert into username values ('" + isname +
"', '" + mac_level + "', '" + conf_level + "', '" + location +
"', 'y')";
           out.println("</a>");
           out.println("<a name=\"runquery\">");

           if (query != null)
           {
          try
                {
                     Class.forName("org.postgresql.Driver");
```

```
                    System.out.println(“Driver loaded”);
                    String
url=“jdbc:postgresql://localhost/IAcontrols”;
                    String user = “postgres”;
                    String pwd = “postgres”;
                    Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                    System.out.println(“Database Connect ok”);
                    Statement
query_stmt=DB_mobile_conn.createStatement();
                    ResultSet
query_rs=query_stmt.executeQuery(query);
                    ResultSetMetaData rsmd =
query_rs.getMetaData();
                    int queryColCount = rsmd.getColumnCount();
                    out.println(“<table border=\”1\”>“);

                    String colName=“”;
                    for ( int i = 1; i <= queryColCount; i++)
                    {
                         colName= colName +
“<td>“+rsmd.getColumnName(i)+”</td>“;
                    }

                    out.println(“<tr>“+colName + “</tr>“ );
                    while (query_rs.next())
                    {
                         sc.log(“Column Returned”);
                         String row = “”;
                         for (int col=1; col <=
queryColCount;col++)
                         {
                              row = row + “<td>“+
query_rs.getString(col)+”</td>“ ;
                         }
                         out.println(“<tr>“+row +”</tr>“ );
                         //out.println(“<tr>“ + row +
“<td>view</td></tr>“);
                    }
                    out.println(“</table>“);
                    query_rs.close();
                    query_stmt.close();

               } catch (Exception exp)
               {
                    System.out.println(“Query exception = “
+exp+ “\n”);
                    System.out.println(“Exception = “ +exp);
          }
       }
```

```
        out.println(“</a>“);
           out.println(“<br>“);
        out.println(“<input type=submit value=“ + “\”Register the
System\”“ + “>“);
        out.println(“</form>“);
        out.print(“<form action=\””);
        out.print(“UserHome\” “);
        out.println(“<br>“);
           out.println(“<input type=submit value=“+”\”Cancel\”“ +
“>“);
           out.println(“</form>“);
        out.println(“</body>“);
        out.println(“</html>“);
    }

    public void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

D. INFORMATION SYSTEM CONTROLS

http://localhost/examples/servlets/servlet/ISControls

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ISControls extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = “<!DOCTYPE HTML PUBLIC \”-//W3C//DTD
HTML 4.0 “ + “Transitional//EN\”>\n”;

        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();
        out.println(“<html>“);
```

```
        out.println(“<head>“);
        out.println(“<title>Continuous Monitoring Program: System
Controls</title>“);
        out.println(“</head>“);
        out.println(“<body style=\”background-
color:lightblue\”>“);
           out.println(“<h3 align=\”center\”>“);
        out.println(“IA Controls for this Information
System</h3>“);
           out.println(“<p align=\”right\”>“);
           out.println(“<a
href=\”http://localhost/examples/servlets/servlet/LogOutPage\”>Si
gn Out</a>“);
           out.println(“</p>“);

           String query = “select mac_level, conf_level from
username where inuse like ‘y’”;
           String query2 = “select A.* from ia_controls A, mac_”;

           if (query != null)
           {
          try
                {
                      Class.forName(“org.postgresql.Driver”);
                      String
url=“jdbc:postgresql://localhost/IAcontrols”;
                      String user = “postgres”;
                      String pwd = “postgres”;
                      Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                      Statement
query_stmt=DB_mobile_conn.createStatement();
                      ResultSet
query_rs=query_stmt.executeQuery(query);

                      while (query_rs.next())
                      {
                            String mac_level =
query_rs.getString(1);
                            String conf_level =
query_rs.getString(2);

                            out.println(“Below are the basic
controls for a “ + mac_level + “ “ + conf_level +”
system:<br><br>“ );

                            if
((mac_level.compareToIgnoreCase(“MAC I”)) == 0) query2 = query2 +
“i_”;
```

```
                              else if
((mac_level.compareToIgnoreCase("MAC II")) == 0) query2 = query2
+ "ii_";
                              else if
((mac_level.compareToIgnoreCase("MAC III")) == 0) query2 = query2
+ "iii_";
                              else query2 = null;

                              if (query2 != null)
                              {
                                   if
((conf_level.compareToIgnoreCase("Classified")) == 0) query2 =
query2 + "classified";
                                   else if
((conf_level.compareToIgnoreCase("Sensitive")) == 0) query2 =
query2 + "sensitive";
                                   else if
((conf_level.compareToIgnoreCase("Public")) == 0) query2 = query2
+ "public";
                                   else query2 = null;
                              }
                         } //end while

                         query_rs.close();
                         query_stmt.close();

                         query2 = query2 + " M where
A.control_number = M.control_number";

                         if (query2 != null)
                         {
                              Statement
query_stmt2=DB_mobile_conn.createStatement();
                              ResultSet
query_rs2=query_stmt2.executeQuery(query2);
                              ResultSetMetaData rsmd =
query_rs2.getMetaData();
                              int queryColCount =
rsmd.getColumnCount();
                              out.println("<table border=\"1\">");

                              String colName="";
                              colName = colName + "<td>Control
Number</td>" + "<td>Control Name</td>" + "<td>IA Service</td>";
                              colName = colName + "<td>Impact
Code</td>" + "<td>Subject Area</td>" + "<td>Description</td>";
                              out.println("<tr>"+colName +"</tr>" );

                              while (query_rs2.next())
                              {
```

```
                                    sc.log(“Column Returned”);
                                    String row = “”;
                                    for (int col=1; col <=
queryColCount;col++)
                                    {
                                         row = row + “<td>“+
query_rs2.getString(col)+”</td>“ ;
                                    }
                                    out.println(“<tr>“+row +”</tr>“
);
                               }
                               out.println(“</table>“);

                               query_rs2.close();
                               query_stmt2.close();

                          }

                    } catch (Exception exp)
                    {
                          System.out.println(“Query exception = “
+exp+ “\n”);
                          System.out.println(“Exception = “ +exp);
                     }
              } else
                 {
                  out.println(“No Query, Please enter “);
              }
              out.println(“<P>“);

                 out.println(“<br>Available actions on this Information
System: <br>“);

                 out.println(“<table border=\”1\” width=\”90%\”>“);
                 out.println(“<tr>“);
                 out.print(“<td width=\”30%\”>“);
                 out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/AppStatus’\”>“);
                 out.println(“Edit Applicability Status of
Controls</button></td>“);
                 out.print(“<td width=\”30%\”>“);
                 out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/ImpStatus’\”>“);
                 out.println(“Edit Implementation Status of
Controls</button></td>“);
                 out.print(“<td width=\”30%\”>“);
```

```
            out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/CompStatus’\”>“);
            out.println(“Edit Compliance Status of
Controls</button></td>“);
            out.println(“</tr>“);
            out.println(“<tr>“);
            out.print(“<td width=\”30%\”>“);
            out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/ControlStatus’\”>“);
            out.println(“View Status of all
Controls</button></td>“);
            out.print(“<td width=\”30%\”>“);
            out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/AddControl’\”>“);
            out.println(“Add a Control</button></td>“);
            out.print(“<td width=\”30%\”>“);
            out.print(“<button
onClick=\”window.location=‘http://localhost/examples/servlets/ser
vlet/RemoveControl’\”>“);
            out.println(“Remove a Control</button></td>“);
            out.println(“</tr>“);
            out.println(“</table>“);

            out.println(“<p align=\”left\”>“);
            out.println(“<a
href=\”http://localhost/examples/servlets/servlet/ISHome\”>Return
to IS Home</a>“);
            out.println(“</p>“);
            out.println(“<p align=\”left\”>“);
            out.println(“<a
href=\”http://localhost/examples/servlets/servlet/UserHome\”>Retu
rn to User Home</a>“);
            out.println(“</p>“);

        out.println(“</body>“);
        out.println(“</html>“);
    }

    public void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

E. VIEW SCANS

http://localhost/examples/servlets/servlet/ViewScans

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ViewScans extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = “<!DOCTYPE HTML PUBLIC \”-//W3C//DTD
HTML 4.0 “ + “Transitional//EN\”>\n”;

        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();
        out.println(“<html>“);
        out.println(“<head>“);
        out.println(“<title>Continuous Monitoring Program: View
Scans</title>“);
        out.println(“</head>“);
        out.println(“<body style=\”background-
color:lightblue\”>“);
           out.println(“<h3 align=\”center\”>“);
        out.println(“View Scans</h3>“);
           out.println(“<p align=\”right\”>“);
           out.println(“<a
href=\”http://localhost/examples/servlets/servlet/LogOutPage\”>Si
gn Out</a>“);
           out.println(“</p>“);
        out.println(“All scans conducted by UserName are
below:<br>“);

           String query = “select * from userscans”;

           if (query != null)
           {
         try
               {
                     Class.forName(“org.postgresql.Driver”);
                     String
url=“jdbc:postgresql://localhost/IAcontrols”;
                     String user = “postgres”;
                     String pwd = “postgres”;
```

```
                    Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                    Statement
query_stmt=DB_mobile_conn.createStatement();
                    ResultSet
query_rs=query_stmt.executeQuery(query);
                    ResultSetMetaData rsmd =
query_rs.getMetaData();
                    int queryColCount = rsmd.getColumnCount();
                    out.println("<table border=\"1\">");

                    String colName="";
                    colName = colName + "<td>Scan
ID</td>"+"<td>IS Name</td>" + "<td>Person Who Conducted the
Scan</td>";
                    colName = colName + "<td>Date</td>" +
"<td>Scan Type</td>" + "<td>File name</td>";
                    out.println("<tr>"+colName + "</tr>" );

                    while (query_rs.next())
                    {
                         sc.log("Column Returned");
                         String row = "";
                         for (int col=1; col <=
queryColCount;col++)
                         {
                              row = row + "<td>"+
query_rs.getString(col)+"</td>" ;
                         }
                         out.println("<tr>"+row +"</tr>" );
                    }
                    out.println("</table>");

                    query_rs.close();
                    query_stmt.close();

               } catch (Exception exp)
               {
                    out.println("Query exception = " +exp+
"\n");
                    System.out.println("Exception = " +exp);
             }
        } else
           {
            out.println("No Query, Please enter ");
        }
        out.println("<P>");
        out.print("<form action=\"");
        out.print("UploadScan\" ");
           out.println("<br>");
```

```
        out.println("<input type=submit value=" + "\"Upload a New
Scan\"" + ">");
           out.println("</form>");
        out.print("<form action=\"");
        out.print("UserHome\" ");
           out.println("<br>");
        out.println("<input type=submit value=" + "\"Return to
User Home\"" + ">");
           out.println("</form>");
        out.println("</body>");
        out.println("</html>");
    }

    public void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

F. INFORMATION SYSTEM SCANS

http://localhost/examples/servlets/servlet/ISScans

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ISScans extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD
HTML 4.0 " + "Transitional//EN\">\n";

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Continuous Monitoring Program: Scans
for this IS</title>");
        out.println("</head>");
```

```
        out.println("<body style=\"background-
color:lightblue\">");
           out.println("<h3 align=\"center\">");
        out.println("Scans for this Information System</h3>");
           out.println("<p align=\"right\">");
           out.println("<a
href=\"http://localhost/examples/servlets/servlet/LogOutPage\">Si
gn Out</a>");
           out.println("</p>");

           String query = "select * from userscans where
isname=(select isname from username where inuse='y')";

           if (query != null)
           {
          try
                {
                      Class.forName("org.postgresql.Driver");
                      String
url="jdbc:postgresql://localhost/IAcontrols";
                      String user = "postgres";
                      String pwd = "postgres";
                      Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                      Statement
query_stmt=DB_mobile_conn.createStatement();
                      ResultSet
query_rs=query_stmt.executeQuery(query);
                      ResultSetMetaData rsmd =
query_rs.getMetaData();
                      int queryColCount = rsmd.getColumnCount();
                      out.println("<table border=\"1\">");

                      String colName="";
                      colName = colName + "<td>Scan
ID</td>"+"<td>IS Name</td>" + "<td>Person Who Conducted the
Scan</td>";
                      colName = colName + "<td>Date</td>" +
"<td>Scan Type</td>" + "<td>File name</td>";
                      out.println("<tr>"+colName + "</tr>" );

                      while (query_rs.next())
                      {
                            sc.log("Column Returned");
                            String row = "";
                            for (int col=1; col <=
queryColCount;col++)
                            {
                                  row = row + "<td>"+
query_rs.getString(col)+"</td>" ;
```

```
                        }
                        out.println("<tr>"+row +"</tr>" );
                    }
                    out.println("</table>");

                    query_rs.close();
                    query_stmt.close();

                } catch (Exception exp)
                {
                    out.println("Query exception = " +exp+
"\n");
                    System.out.println("Exception = " +exp);
                }
        } else
           {
            out.println("No Query, Please enter ");
        }
        out.println("<P>");
        out.print("<form action=\"");
        out.print("UploadScan\" ");
           out.println("<br>");
        out.println("<input type=submit value=" + "\"Upload a New
Scan\"" + ">");
           out.println("</form>");
        out.print("<form action=\"");
        out.print("UserHome\" ");
           out.println("<br>");
        out.println("<input type=submit value=" + "\"Return to
User Home\"" + ">");
           out.println("</form>");
        out.println("</body>");
        out.println("</html>");
    }

    public void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

G. APPLICABILITY STATUS

http://localhost/examples/servlets/servlet/AppStatus

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class AppStatus extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = “<!DOCTYPE HTML PUBLIC \”-//W3C//DTD
HTML 4.0 “ + “Transitional//EN\”>\n”;

        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();
        out.println(“<html>“);
        out.println(“<head>“);
        out.println(“<title>Continuous Monitoring Program:
Applicability Status</title>“);
        out.println(“</head>“);
        out.println(“<body style=\”background-
color:lightblue\”>“);
           out.println(“<h3 align=\”center\”>“);
        out.println(“Applicability Status of Controls</h3>“);
           out.println(“<p align=\”right\”>“);
           out.println(“<a
href=\”http://localhost/examples/servlets/servlet/LogOutPage\”>Si
gn Out</a>“);
           out.println(“</p>“);
           out.println(“Label each control as Inherited,
Applicable, or Not Applicable.<br><br>“);

           String query = “select * from is1234 order by
control_number”;

           if (query != null)
           {
                 try
                 {
                       Class.forName(“org.postgresql.Driver”);
                       System.out.println(“Driver loaded”);
                       String
url=“jdbc:postgresql://localhost/IAcontrols”;
                       String user = “postgres”;
                       String pwd = “postgres”;
                       Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
```

```
                    System.out.println(“Database Connect ok”);

                    Statement
query_stmt=DB_mobile_conn.createStatement();
                    ResultSet
query_rs=query_stmt.executeQuery(query);
                    ResultSetMetaData rsmd =
query_rs.getMetaData();
                    int queryColCount = rsmd.getColumnCount();

                    out.println(“<form
action=\”AppStatus#change\” method=POST>“);
                    out.println(“<table border=\”1\”>“);

                    String colName=“”;
                    for ( int i = 1; i < queryColCount-1; i++)
                    {
                         colName= colName +
“<td>“+rsmd.getColumnName(i)+”</td>“;
                    }
                    out.println(“<tr>“+colName + “</tr>“ );

                    while (query_rs.next())
                    {
                         sc.log(“Column Returned”);
                         String row = “”;
                         for (int col=1; col < queryColCount-
2;col++)
                         {
                              row = row + “<td>“+
query_rs.getString(col)+”</td>“ ;
                         }
                         out.println(“<tr>“+row + “<td>“);

                         out.println(“<form>“);
                         out.println(“<select
name=applicability>“);
                         out.println(“<option
values=“+”Applicable”+”>Applicable</option>“);
                         out.println(“<option
values=“+”Inherited”+”>Inherited</option>“);
                         out.println(“<option values=“+”Not
Applicable”+”>Not Applicable</option>“);
                         out.println(“</select>“);
                         out.println(“<br>“);
                         out.println(“</form>“);

                         out.println(“</td></tr>“);
                    }
                    out.println(“</table>“);
```

```
                    out.println("<input type=submit value=" +
"\"Save changes\"" + ">");
                    out.println("</form>");

                    query_rs.close();
                    query_stmt.close();

               } catch (Exception exp)
               {
                    System.out.println("Query exception = "
+exp+ "\n");
           }
          } //end of query

          out.println("<P>");
          out.println("<p align=\"left\">");
          out.println("<a
href=\"http://localhost/examples/servlets/servlet/ISControls\">Re
turn to IS Controls</a>");
          out.println("</p>");
          out.println("<p align=\"left\">");
          out.println("<a
href=\"http://localhost/examples/servlets/servlet/ISHome\">Return
to IS Home</a>");
          out.println("</p>");
          out.println("<p align=\"left\">");
          out.println("<a
href=\"http://localhost/examples/servlets/servlet/UserHome\">Retu
rn to User Home</a>");
          out.println("</p>");
        out.println("</body>");
        out.println("</html>");
    }

         public void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

H. IMPLEMENTATION STATUS

http://localhost/examples/servlets/servlet/ImpStatus

```
import java.sql.*;
import java.io.*;
import java.lang.*;
```

```
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ImpStatus extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = “<!DOCTYPE HTML PUBLIC \”-//W3C//DTD
HTML 4.0 “ + “Transitional//EN\”>\n”;

        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();
        out.println(“<html>“);
        out.println(“<head>“);
        out.println(“<title>Continuous Monitoring Program:
Implementation Status</title>“);
        out.println(“</head>“);
        out.println(“<body style=\”background-
color:lightblue\”>“);
           out.println(“<h3 align=\”center\”>“);
        out.println(“Implementation Status of Controls</h3>“);
           out.println(“<p align=\”right\”>“);
           out.println(“<a
href=\”http://localhost/examples/servlets/servlet/LogOutPage\”>Si
gn Out</a>“);
           out.println(“</p>“);
           out.println(“Label each control as Implemented or Not
Implemented.<br><br>“);

           String query = “select * from is1234 order by
control_number”;

           if (query != null)
           {
                try
                {
                     Class.forName(“org.postgresql.Driver”);
                     System.out.println(“Driver loaded”);
                     String
url=“jdbc:postgresql://localhost/IAcontrols”;
                     String user = “postgres”;
                     String pwd = “postgres”;
                     Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                     System.out.println(“Database Connect ok”);
```

```
                    Statement
query_stmt=DB_mobile_conn.createStatement();
                    ResultSet
query_rs=query_stmt.executeQuery(query);
                    ResultSetMetaData rsmd =
query_rs.getMetaData();
                    int queryColCount = rsmd.getColumnCount();

                    out.println("<form
action=\"ImpStatus#change\" method=POST>");
                    out.println("<table border=\"1\">");

                    String colName="";
                    for ( int i = 1; i < queryColCount; i++)
                    {
                         colName= colName +
"<td>"+rsmd.getColumnName(i)+"</td>";
                    }
                    out.println("<tr>"+colName + "</tr>" );

                    while (query_rs.next())
                    {
                         sc.log("Column Returned");
                         String row = "";
                         for (int col=1; col < queryColCount-
1;col++)
                         {
                              row = row + "<td>"+
query_rs.getString(col)+"</td>" ;
                         }
                         out.println("<tr>"+row + "<td>");
                         if (query_rs.getString(queryColCount-
2).equals("Applicable"))
                         {
                              out.println("<form>");
                              out.println("<select
name=implementation>");
                              out.println("<option
values="+"Inherited"+">Implemented</option>");
                              out.println("<option
values="+"Not Applicable"+">Not Implemented</option>");
                              out.println("</select>");
                              out.println("<br>");
                              out.println("</form>");
                         }
                         else
                         {

     out.println(query_rs.getString(queryColCount-2));
                         }
```

```
                              out.println("</td></tr>");
                         }
                         out.println("</table>");
                         out.println("<input type=submit value=" +
"\"Save changes\"" + ">");
                         out.println("</form>");

                         query_rs.close();
                         query_stmt.close();

                    } catch (Exception exp)
                    {
                         System.out.println("Query exception = "
+exp+ "\n");
                     }
               } //end of query


               out.println("<P>");
               out.println("<p align=\"left\">");
               out.println("<a
href=\"http://localhost/examples/servlets/servlet/ISControls\">Re
turn to IS Controls</a>");
               out.println("</p>");
               out.println("<p align=\"left\">");
               out.println("<a
href=\"http://localhost/examples/servlets/servlet/ISHome\">Return
to IS Home</a>");
               out.println("</p>");
               out.println("<p align=\"left\">");
               out.println("<a
href=\"http://localhost/examples/servlets/servlet/UserHome\">Retu
rn to User Home</a>");
               out.println("</p>");
            out.println("</body>");
            out.println("</html>");
        }

             public void doPost(HttpServletRequest request,
HttpServletResponse response)
        throws IOException, ServletException
        {
            doGet(request, response);
        }
}
```

I. COMPLIANCE STATUS

http://localhost/examples/servlets/servlet/CompStatus

```
import java.sql.*;
import java.io.*;
import java.lang.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class CompStatus extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = “<!DOCTYPE HTML PUBLIC \”-//W3C//DTD
HTML 4.0 “ + “Transitional//EN\”>\n”;

        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();
        out.println(“<html>“);
        out.println(“<head>“);
        out.println(“<title>Continuous Monitoring Program:
Compliance Status</title>“);
        out.println(“</head>“);
        out.println(“<body style=\”background-
color:lightblue\”>“);
           out.println(“<h3 align=\”center\”>“);
        out.println(“Compliance Status of Controls</h3>“);
           out.println(“<p align=\”right\”>“);
           out.println(“<a
href=\”http://localhost/examples/servlets/servlet/LogOutPage\”>Si
gn Out</a>“);
           out.println(“</p>“);
           out.println(“Label each control as Compliant or Not
Compliant.<br><br>“);

           String query = “select * from is1234 order by
control_number”;

           if (query != null)
           {
                try
                {
                     Class.forName(“org.postgresql.Driver”);
                     System.out.println(“Driver loaded”);
```

```
                        String
url="jdbc:postgresql://localhost/IAcontrols";
                        String user = "postgres";
                        String pwd = "postgres";
                        Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                        System.out.println("Database Connect ok");

                        Statement
query_stmt=DB_mobile_conn.createStatement();
                        ResultSet
query_rs=query_stmt.executeQuery(query);
                        ResultSetMetaData rsmd =
query_rs.getMetaData();
                        int queryColCount = rsmd.getColumnCount();

                        out.println("<form
action=\"CompStatus#change\" method=POST>");
                        out.println("<table border=\"1\">");

                        String colName="";
                        for ( int i = 1; i <= queryColCount; i++)
                        {
                              colName= colName +
"<td>"+rsmd.getColumnName(i)+"</td>";
                        }
                        out.println("<tr>"+colName + "</tr>" );

                        while (query_rs.next())
                        {
                              sc.log("Column Returned");
                              String row = "";
                              for (int col=1; col <
queryColCount;col++)
                              {
                                    row = row + "<td>"+
query_rs.getString(col)+"</td>" ;
                              }
                              out.println("<tr>"+row + "<td>");
                              if (query_rs.getString(queryColCount-
2).equals("Applicable"))
                              {
                                    out.println("<form>");
                                    out.println("<select
name=comliance>");
                                    out.println("<option
values="+"Compliant"+">Compliant</option>");
                                    out.println("<option
values="+"Not Compliant"+">Not Compliant</option>");
                                    out.println("</select>");
```

```
                                    out.println(“<br>“);
                                    out.println(“</form>“);
                               }
                               else
                               {

     out.println(query_rs.getString(queryColCount-2));
                               }
                               out.println(“</td></tr>“);
                          }
                          out.println(“</table>“);
                          out.println(“<input type=submit value=“ +
“\”Save changes\”“ + “>“);
                          out.println(“</form>“);

                          query_rs.close();
                          query_stmt.close();

                    } catch (Exception exp)
                    {
                          System.out.println(“Query exception = “
+exp+ “\n”);
                     }
               } //end of query

               out.println(“<P>“);
               out.println(“<p align=\”left\”>“);
               out.println(“<a
href=\”http://localhost/examples/servlets/servlet/ISControls\”>Re
turn to IS Controls</a>“);
               out.println(“</p>“);
               out.println(“<p align=\”left\”>“);
               out.println(“<a
href=\”http://localhost/examples/servlets/servlet/ISHome\”>Return
to IS Home</a>“);
               out.println(“</p>“);
               out.println(“<p align=\”left\”>“);
               out.println(“<a
href=\”http://localhost/examples/servlets/servlet/UserHome\”>Retu
rn to User Home</a>“);
               out.println(“</p>“);
            out.println(“</body>“);
            out.println(“</html>“);
        }

             public void doPost(HttpServletRequest request,
HttpServletResponse response)
        throws IOException, ServletException
        {
            doGet(request, response);
```

```
    }
}
```

J. SYSTEM RISK

http://localhost/examples/servlets/servlet/SystemRisk

```
import java.sql.*;
import java.lang.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SystemRisk extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
           ServletContext sc = getServletContext();
           String docType = "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD
HTML 4.0 " + "Transitional//EN\">\n";

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Continuous Monitoring Program: System
Risk</title>");
        out.println("</head>");
        out.println("<body style=\"background-
color:lightblue\">");
        out.println("<h3 align=\"center\">View/Accept System
Risk</h3>");
           out.println("<p align=\"right\">");
           out.println("<a
href=\"http://localhost/examples/servlets/servlet/LogOutPage\">Si
gn Out</a>");
           out.println("</p>");
           int highVal=0;
           int mediumVal=0;
           int lowVal=0;

           String query = "select count(*) from is1234 where
impact_code='High' and compliance = 'Not Compliant'";
           String query1 = "select count(*) from is1234 where
impact_code='Medium' and compliance = 'Not Compliant'";
```

```
          String query2 = “select count(*) from is1234 where
impact_code=‘Low’ and compliance = ‘Not Compliant’”;

          if (query != null && query1 != null && query2 != null)
          {
         try
               {
                     Class.forName(“org.postgresql.Driver”);
                     String
url=“jdbc:postgresql://localhost/IAcontrols”;
                     String user = “postgres”;
                     String pwd = “postgres”;
                     Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                     Statement
query_stmt=DB_mobile_conn.createStatement();

                     out.println(“Number of non compliant
controls by Impact Code: <br>“);
                     out.println(“<table border=\”1\”
width=\”60%\”>“);
                     out.println(“<tr><td>Impact Code</td>“ +
“<td>Number of Non Compliant Controls</td></tr>“);

                     ResultSet
query_rs=query_stmt.executeQuery(query);
                     while (query_rs.next())
                     {
                           highVal =
Integer.parseInt(query_rs.getString(1));
                           out.println(“<tr><td><font
color=\”red\”>High</font></td>“);
                           out.println(“<td><font color=\”red\”>“
+ query_rs.getString(1) + “</font></td></tr>“);
                     }
                     query_rs.close();

                     ResultSet
query_rs1=query_stmt.executeQuery(query1);
                     while (query_rs1.next())
                     {
                           mediumVal =
Integer.parseInt(query_rs1.getString(1));
                           out.println(“<tr><td><font
color=\”orange\”>Medium</font></td>“);
                           out.println(“<td><font
color=\”orange\”>“ + query_rs1.getString(1) +
“</font></td></tr>“);
                     }
                     query_rs1.close();
```

```
                    ResultSet
query_rs2=query_stmt.executeQuery(query2);
                    while (query_rs2.next())
                    {
                         lowVal =
Integer.parseInt(query_rs2.getString(1));
                         out.println("<tr><td><font
color=\"green\">Low</font></td>");
                         out.println("<td><font
color=\"green\">" + query_rs2.getString(1) +
"</font></td></tr>");
                    }
                    query_rs2.close();

                    out.println("</table>");
                    query_stmt.close();

               } catch (Exception exp)
               {
                    out.println("Query exception = " +exp+
"\n");
                    System.out.println("Exception = " +exp);
                }
        } else
           {
            out.println("No Query, Please enter ");
        }

           out.println("<P>");
           int score = highVal*10 + mediumVal*5 + lowVal*1;
        out.println("<br>");
           out.println("<p><b>");
           out.println("The Overall System Risk Score is " +
score + "%");
           out.println("</b></p>");
           out.println("<br>");

           out.print("<button
onClick=\"window.location='http://localhost/examples/servlets/ser
vlet/ISHome'\">");
           out.println("Accept System Risk</button></td>");
           out.println("<br>");
           out.println("<br>");

           out.println("<p align=\"left\">");
           out.println("<a
href=\"http://localhost/examples/servlets/servlet/ISHome\">Return
to IS Home</a>");
           out.println("</p>");
```

```
            out.println(“<p align=\”left\”>“);
            out.println(“<a
href=\”http://localhost/examples/servlets/servlet/UserHome\”>Retu
rn to User Home</a>“);
            out.println(“</p>“);

            out.println(“</body>“);
        out.println(“</html>“);
    }

    public void doPost(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
        doGet(request, response);
    }
}
```

K. RETIRE SYSTEM

http://localhost/examples/servlets/servlet/RetireSystem

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class RetireSystem extends HttpServlet {

    public void doGet(HttpServletRequest request,
HttpServletResponse response)
    throws IOException, ServletException
    {
            ServletContext sc = getServletContext();
            String docType = “<!DOCTYPE HTML PUBLIC \”-//W3C//DTD
HTML 4.0 “ + “Transitional//EN\”>\n”;

        response.setContentType(“text/html”);
        PrintWriter out = response.getWriter();
        out.println(“<html>“);
        out.println(“<head>“);
        out.println(“<title>Continuous Monitoring Program: Retire
IS</title>“);
        out.println(“</head>“);
        out.println(“<body style=\”background-
color:lightblue\”>“);
            out.print(“<h3 align=\”center\”>“);
```

```
        out.println(“Retire an Information System</h3>“);
           out.println(“<p align=\”right\”>“);
           out.println(“<a
href=\”http://localhost/examples/servlets/servlet/LogOutPage\”>Si
gn Out</a>“);
           out.println(“</p>“);
        out.println(“Are you sure you wish to retire the system
below?<br>“);

           String query = “select * from username where inuse
like ‘y’”;
           if (query != null)
           {
          try
                {
                      Class.forName(“org.postgresql.Driver”);
                      String
url=“jdbc:postgresql://localhost/IAcontrols”;
                      String user = “postgres”;
                      String pwd = “postgres”;
                      Connection DB_mobile_conn =
DriverManager.getConnection(url,user,pwd);
                      Statement
query_stmt=DB_mobile_conn.createStatement();
                      ResultSet
query_rs=query_stmt.executeQuery(query);
                      ResultSetMetaData rsmd =
query_rs.getMetaData();
                      int queryColCount = rsmd.getColumnCount();
                      out.println(“<table border=\”1\”>“);

                      while (query_rs.next())
                      {
                            sc.log(“Column Returned”);
                            String row = ““;
                            int col=1;
                            row = “<td>IS Name: </td>“ + “<td>“+
query_rs.getString(col)+”</td>“;
                            out.println(“<tr>“+row + “</tr>“);
                            col++;
                            row = “<td>IS Type: </td>“ + “<td>“+
query_rs.getString(col)+”</td>“;
                            out.println(“<tr>“+row + “</tr>“);
                            col++;
                            row = “<td>Designated Accrediting
Authority (DAA):</td>“ + “<td>“+ query_rs.getString(col)+”</td>“;
                            out.println(“<tr>“+row + “</tr>“);
                            col++;
                            row = “<td>Certifying Authority (CA):
</td>“ + “<td>“+ query_rs.getString(col)+”</td>“;
```

```
                              out.println("<tr>"+row + "</tr>");
                              col++;
                              row = "<td>MAC Level: </td>" + "<td>"+
query_rs.getString(col)+"</td>";
                              out.println("<tr>"+row + "</tr>");
                              col++;
                              row = "<td>Confidentiality Level:
</td>" + "<td>"+ query_rs.getString(col)+"</td>";
                              out.println("<tr>"+row + "</tr>");
                              col++;
                              row = "<td>Site/Command Name: </td>" +
"<td>"+ query_rs.getString(col)+"</td>";
                              out.println("<tr>"+row + "</tr>");
                         }
                         out.println("</table>");

                         query_rs.close();
                         query_stmt.close();

                    } catch (Exception exp)
                    {
                         System.out.println("Query exception = "
+exp+ "\n");
                         System.out.println("Exception = " +exp);
                     }
            }
               out.println("<P>");
               out.print("<form action=\"");
               out.print("SystemRetired\" ");
               out.println("<br>");
               out.println("<input type=submit value=" + "\"Retire
the IS\"" + ">");
               out.println("</form>");
               out.print("<form action=\"");
            out.print("UserHome\" ");
               out.println("<br>");
            out.println("<input type=submit value=" + "\"Cancel\"" +
">");
               out.println("</form>");
            out.println("</body>");
            out.println("</html>");
        }
        public void doPost(HttpServletRequest request,
HttpServletResponse response)
        throws IOException, ServletException
        {
            doGet(request, response);
        }
}
```

LIST OF REFERENCES

[1] Richard K. Betts, *Conflict after the Cold War: Arguments on Causes of War and Peace*, 3rd Edition. San Francisco: Pearson Education Inc., 2008, page 430.

[2] Committee on National Security Systems (CNSS), (2010, April 26). *Instruction 4009, National Information Assurance Glossary*. [Online]. Available: http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf.

[3] United States Department of Justice, Office of Programs, (2002, December 17). *Public Law 107–347, E-Government Act* [includes Federal information Security Management Act (FISMA)]. [Online]. Available: http://it.ojp.gov/default.aspx?area=privacy&page=1287#contentTop.

[4] Office of Management and Budget, Management of Federal Information Resources, (2000, November). *Circular A-130, Appendix III, Transmittal Memorandum #4, Management of Federal Information Resources*. [Online]. Available: http://www.whitehouse.gov/omb/circulars_a130_a130trans4.

[5] United States Department of Defense, (2007, November 28). *Instruction 8510.01, DoD Information Assurance Certification and Accreditation Process (DIACAP)*. [Online]. Available: http://www.dtic.mil/whs/directives/corres/pdf/851001p.pdf.

[6] United States Department of Defense, (2003, February 6). *Instruction 8500.2, Information Assurance (IA) Implementation*. [Online]. Available: http://www.dtic.mil/whs/directives/corres/pdf/850002p.pdf.

[7] United States Department of Defense,(2002, October 24). *Directive 8500.01E, Information Assurance (IA)*. [Online]. Available: http://www.dtic.mil/whs/directives/corres/pdf/850001p.pdf.

[8] United States Department of Defense, (2002, September 19). *Directive 8100.1, Global Information Grid (GIG) Overarching Policy*. [Online]. Available: http://biotech.law.lsu.edu/blaw/dodd/corres/pdf/d81001_091902/d81001p.pdf.

[9] Information Assurance Training Center, U.S. Army Signal Center, "Information Assurance Fundamentals (IAF) Training," September 2011, https://ia.signal.army.mil/IAF/IASOLesson11.asp.

[10] U.S. Navy, "DIACAP Knowledge Service," September 2011, https://diacap.iaportal.navy.mil/.

[11] *Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach*, National Institute of Standards and Technology, Special Publication 800–37 Revision 1, February 2010.

[12] National Institute of Standards and Technology, "Applying the Risk Management Framework to Federal Information Systems course," September 2011, http://csrc.nist.gov/groups/SMA/fisma/Risk-Management-Framework/rmf-training/index.html.

[13] *Standards for Security Categorization of Federal Information and Information Systems*, Federal Information Processing Standards Publication 199, February 2004.

[14] *Guide for Mapping Types of information and information Systems to Security Categories*, National Institute of Standards and Technology, Special Publication 800–60 Volume 1 Revision 1, August 2008.

[15] *Minimum Security Requirements for Federal Information and Information Systems*, Federal Information Processing Standards Publication 200, March 2006.

[16] *Recommended Security Controls for Federal Information Systems and Organizations*, National Institute of Standards and Technology Special Publication 800–53 Revision 3, August 2009.

[17] *Guide for Assessing the Security Controls in Federal Information Systems and Organizations, Building Effective Security Assessment Plans*, National Institute of Standards and Technology Special Publication 800–53A Revision 1, June 2010.

[18] Department of State, “Continuous Certification and Accreditation (C&A) Strategy,” September 2011, http://www.state.gov/documents/organization/156898.pdf

[19] Navy ODAA and NAVINTEL DAO. (12 April 2010). *Navy DoD & SCI Systems Transformational Certification & Accreditation Process Plan Version 3.0*.

[20] Defense Information Systems Agency (DISA), Network Services Directorate (NS) Connection Approval Division (NSC). (2010, May). *Connection Process Guide Version 3*. [Online]. Available: http://www.disa.mil/connect/library/files/disn_cap_05012010_v3.pdf.

[21] United States of America, Chairman of the Joint Chiefs of Staff, (2008, July 9). *Instruction 6211.02C, Defense Information System Network (DISN): Policy and Responsibilities*. [Online]. Available: http://www.dtic.mil/cjcs_directives/cdata/unlimit/6211_02.pdf.

[22] PostgreSQL, “PostgreSQL: The world’s most advanced open source database,” PostgreSQL Global Development Group, September 2011, http://www.postgresql.org/.

[23] pgAdmin, “pgAdmin: PostgreSQL administration and management tools,” September 2011, http://www.pgadmin.org/.

[24] Apache Tomcat, “Tutorial: Configuring & Using Apache Tomcat 6 and Apache Tomcat 7,” coreservlets.com, September 2011, http://www.coreservlets.com/Apache-Tomcat-Tutorial/.

[25] “Apache Tomcat,” Wikipedia. September 2011, http://en.wikipedia.org/wiki/Apache_Tomcat.

[26] Java SE, “Java SE Downloads,” Oracle, September 2011, http://www.oracle.com/technetwork/java/javase/downloads/index.html.

[27] Refsnes Data, “HTML Tutorial,” September 2011, http://www.w3schools.com/html/.

[28] “Database Systems,” class notes for CS3060, Department of Computer Science, Graduate School of Operational & Information Sciences, Naval Postgraduate School, Winter 2011.

[29] Internet Explorer, “Internet Explorer – Web Browser for Microsoft Windows,” Microsoft Corporation, September 2011, http://windows.microsoft.com/en-U.S./Internet-explorer/products/ie/home.

[30] *Managing Information Security Risk: Organization, Mission and Information System View*, National Institute of Standards and Technology Special Publication 800–39, March 2011.

[31] *Information Security Continuous Monitoring for Federal Information Systems and Organizations*, National Institute of Standards and Technology Special Publication 800–137, 16 December 2010.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California